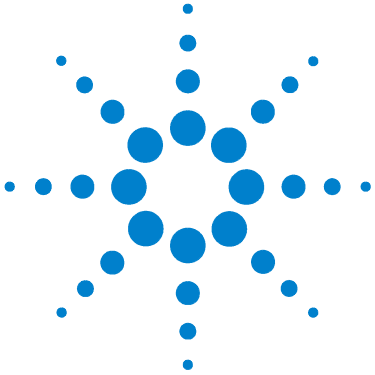


Agilent 85190A
IC-CAP 2004

Nonlinear Device Models
Volume 1

Agilent Technologies

Notices

© Agilent Technologies, Inc. 2000-2004

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

Manual Part Number

85190-90139

Edition

August 2004

Printed in USA

Agilent Technologies, Inc.
395 Page Mill Road
Palo Alto, CA 94304 USA

Acknowledgments

UNIX ® is a registered trademark of the Open Group.

Windows ®, MS Windows ® and Windows NT ® are U.S. registered trademarks of Microsoft Corporation.

Errata

The IC-CAP product may contain references to “HP” or “HPEESOF” such as in file names and directory names. The business entity formerly known as “HP EEsof” is now part of Agilent Technologies and is known as “Agilent EEsof.” To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain “HP” or “HPEESOF” references.

Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as “Commercial computer software” as defined in DFAR 252.227-7014 (June 1995), or as a “commercial item” as defined in FAR 2.101(a) or as “Restricted computer software” as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies’ standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Contents

1 Bipolar Transistor Characterization

Bipolar Device Model 15

Simulators 16

Model Parameters 17

Test Instruments 23

Instrument-to-Device Connections 23

Measuring and Extracting 25

Measurement and Extraction Guidelines 25

Extracting Parameters 29

Extraction Algorithms 33

DC Parameter Extractions 33

Capacitance Parameter Extractions 34

Parasitic DC Parameter Extractions 34

AC Parameter Extractions 35

References 37

2 MOSFET Characterization

UCB MOSFET Model 41

Simulators 42

MOSFET Model Parameters 43

Test Instruments 49

Instrument-to-Device Connections 49

Measuring and Extracting 51

Measurement and Extraction Guidelines 51

Extracting Parameters 53
Simulating 59
Displaying Plots 60
Optimizing 60
Extraction Algorithms 61
Classical Parameter Extractions 61
Narrow-Width Parameter Extractions 61
Short-Channel Parameter Extractions 62
Saturation Parameter Extractions 62
Sidewall and Junction Capacitance Parameter Extractions 62
HSPICE LEVEL 6 MOSFET Model 64
Model Parameters 64
Measurement 66
Extraction and Optimization 67
References 68

3 Using the BSIM Modeling Packages

Key Features of the BSIM3 and BSIM4 Modeling Packages 70
Data Structure inside the BSIM3 and BSIM4 Modeling Packages 72
Files resulting from Measurement and Extraction using the Modeling Packages 73
DC and CV Measurement of MOSFET's for the BSIM3 and BSIM4 Models 77
Project Notes 80
DC Measurement Conditions 81
Temperature Setup 87
Switch Matrix / Wafer Prober 88
DC Transistor DUTs 92
Capacitance DUTs 108
DC Diode DUTs 116

Drain/Source – Bulk Diodes for DC Measurements 121
Options 121

RF Measurement 123
RF Measurement Notes 123
RF Measurement Conditions 123
Deembedding/Pad Structures 126
DUTs 132
RF Measurement Options 136

Extraction of DC/CV Parameters 137
DC Notes 139
DC Information 139
DC Initialize 140
Binning 145
DC Extract 150
Plot Optimizer 158
Configuration of fine-tuning tasks using the BSIM3/4_DC_Extract model 160
DC Display 164
DC HTML 165
DC Options 169
DC Boundaries 172

Extraction of Parameters for the RF Models 176
RF Extract Notes 177
RF Extract Information 177
RF Extract Initialize 177
Model Parameter Sets 179
RF Extract 182
RF Extract Display 186
RF Extract HTML 187
RF Extract Options 187
Circuit Files 189
RF Extract Boundaries 189

4 BSIM3v3 Characterization

What's new inside the BSIM3 Modeling Package: 192

The BSIM3 Model 197

Versions of the BSIM3 Model 198

The Unified I-V Model of BSIM3v3 200

Threshold Voltage 200

Carrier Mobility Reduction 210

Effective Channel Length and Width 212

Drain Current 215

Output Resistance 224

Substrate Current 230

Drain / Bulk and Source / Bulk Diodes 232

Consistency Check of DC measurement data for multiple measured devices 234

Capacitance Model 238

Junction Capacitance 239

Extrinsic Capacitance 244

Intrinsic Capacitance 247

The Overall Capacitance in BSIM3 252

High Frequency Behavior 255

Macro Model for High Frequency Application 255

Single Subcircuit Model for BSIM3v3 RF Transistors 257

Fully Scalable Subcircuit Model for BSIM3v3 RF Transistors 261

Modeling Strategy 266

Temperature Dependence 271

Built-in Temperature Dependencies 271

Temperature Effects 271

Noise Model 279

Conventional Noise Model for MOS Devices 279

BSIM3v3 Noise Model 281

SPICE Model Parameters 282
Main Model Parameters 282
Process Related Parameters 288
Temperature Modeling Parameters 289
Flicker Noise Model Parameters 289
Non-Quasi-Static Model Parameters 290
Model Selection Flags 290
User Definable Parameters 291
Additional Parameters needed for accurate RF modeling 291
Test structures for Deep Submicron CMOS Processes 293
Transistors for DC measurements 293
Drain/Source – Bulk Diodes for DC Measurements 296
Test Structures for CV Measurements 296
Testchips 299
Test Structures for S-parameter Measurements 299
Extraction of Model Parameters 311
Parameter Extraction Sequence 311
Extraction Strategy 312
Binning of Model Parameters 318
Usage of binned models in a simulator 318
Extraction of binned model parameters 322
Implementation into the BSIM3/4 Modeling Packages 324
Importing older version BSIM3v3 Files 328
References 332

5 BSIM4 Characterization

What's new inside the BSIM4 Modeling Package: 337
Basic Effects Modeled in BSIM4 341
Enhanced drain current model 341
RF and high-speed model 341

Key Features of the BSIM4 Modeling Package 342

DC Behavioral Modeling 343
- Threshold Voltage Model 343
- Basic Threshold voltage equation 347
- Gate Direct Tunneling Current Model 353
- Drain Current Model 359
- Unified Mobility Model 361
- Drain/Source Resistance Model 363
- Saturation Region Output Conductance Model 365
- Body Current Model 371
- Stress Effect Modeling 372

CV Modeling 376
- Capacitance Model 376
- Intrinsic Capacitance Modeling 376
- Intrinsic Capacitance Model Equations 379
- Fringing Capacitance Models 379
- Overlap Capacitance Model 379

Structure of the BSIM4 RF Simulation Model 383

Test Structures for Deep Submicron CMOS Processes 399

Tutorials for the BSIM4 Modeling Package 400
- Opening the Tutorial 400

SPICE Model Parameters 406
- New Model Parameters for BSIM4.3.0 406
- Main Model Parameters 408
- Temperature Modeling Parameters 419
- Flicker Noise Model Parameters 420
- High-Speed / RF Model Parameters 421
- Layout-Dependent Parasitics Model Parameters 422
- Model Selection Flags 423

References 425

6 UCB GaAs MESFET Characterization

UCB GaAs MESFET Model 429

Simulators 430

Model Parameters 433

Test Instruments 435

Instrument-to-Device Connections 435

Measuring and Extracting 436

Measurement and Extraction Guidelines 436

Extraction Procedure Overview 439

Parameter Measurement and Extraction 440

Alternate Extraction Method 442

Simulating 444

Displaying Plots 444

Optimizing 444

Extraction Algorithms 446

Inductance and Resistance Extraction 446

DC Parameter Extractions 446

Capacitance Parameter Extractions 446

References 447

7 Curtice GaAs MESFET Characterization

Curtice GaAs MESFET Model 451

Simulators 452

Model Parameters 454

Test Instruments 458

Instrument-to-Device Connections 458

Measuring and Extracting 460

Measurement and Extraction Guidelines 460

Extraction Procedure Overview 463

Parameter Measurement and Extraction 464
Alternate Extraction Method 467
Simulating 469
Displaying Plots 469
Optimizing 470

Extraction Algorithms 471
Inductance and Resistance Extraction 471
DC Parameter Extractions 471
Capacitance Parameter Extractions 472

References 473

8 MOS Model 9 Characterization

MOS Model 9 Model 477

The MM9 Model File 479
Model Parameters 479
Model Variables 482
The extract DUT 489
The quick_ext DUT 497
The dutx DUT 502
Macros 506

Parameter Extraction 511
Data Organization 513
Scaling Rules 513
Device Geometries 514

Optimizing 516
Optimization Transforms and Macros 516

The JUNCAP Model 519
The area, locos, and gate DUTs 519
The analysis DUT 521
Macros 530

General Extraction Methodology 533

References 536

9 Circuit Modeling

Definition of an IC-CAP Circuit 538

IC-CAP Circuit Modeling Operations 539

Defining a Circuit 540

Supported Circuit Components 540

Circuit Measurement 543

Multiple Instrument Names 543

Isolating Circuit Elements for Measurement and Extraction 543

Circuit Parameter Extraction 545

Extracting Transistor Parameters Using Library Functions 545

Extracting Parameters Through Optimization 546

Circuit Simulation 549

Design Optimization 549

Test Circuits 551

Syntax 551

Hierarchical Modeling 553

Circuits Built from Sub-models 553

Functional Circuit Blocks 555

Types of Circuits in IC-CAP 555

Modeling the Reverse Active Region of an NPN Transistor 555

Modeling an Operational Amplifier 557

References 566

10 1/f Noise Extraction Toolkit

Types of Noise 568

The Significance of 1/f Noise 569

1/f (Flicker Noise) Modeling 570

Noise Measurement Setup 573

Parameter Extraction Procedure 575

Low pass filter 576

1/f Noise Toolkit Description 578

System Parts List (Hardware) 578

Software Requirements 578

Opening the 1/f Toolkit 579

System Settings 582

1/f Noise System Settings 582

Notes 582

General Settings 584

Instrument Settings 585

GUI Options 588

Verify DC 590

Measure Noise 595

1/f Noise Extraction with MOS Transistors 614

1/f Noise Function Descriptions 627

References 631

Index

1 Bipolar Transistor Characterization

Bipolar Device Model 15
Simulators 16
Model Parameters 17
Test Instruments 23
Measuring and Extracting 25
Extraction Algorithms 33
References 37

This chapter describes the University of California at Berkeley bipolar transistor model supported in SPICE. Descriptions of model setup, instrument connections, and model parameters are included. Information is included for making DC, capacitance, and high-frequency AC measurements and their corresponding extractions.

The IC-CAP bipolar modeling module provides setups that can be used for general measurement and model extraction for bipolar devices. Two example files are provided for the bipolar model; the example files can also be used as a template for creating custom model configurations.

bjt_npn.mdl extracts parameters for NPN bipolar transistors

bjt_pnp.mdl extracts parameters for PNP bipolar transistors

The IC-CAP system offers the flexibility to modify any measurement or simulation specification.

The model extractions provided are also intended for general bipolar IC processes. If you have another method of extracting specific model parameters, you can do so with the Program

function by writing a function in C and linking it to the function list. For Program function details or for writing user-defined C-language routines, refer to Chapter 9, "Using Transforms and Functions," in the *IC-CAP User's Guide*.

The following shows an example custom extraction for IS and NF:

```
! Extraction for IS & NF
! Note: print statements go to the
!       window that started IC-CAP
print "Example Custom Extraction for IS & NF"
index = 0              ! array index
! pick two low current points
v1 = -ve[index]
WHILE v1 < 0.4         ! get a point near Vbe = 0.4
  index = index + 1
  v1 = -ve[index]
END WHILE
i1 = ic.m[index]
v2 = -ve[index]
WHILE v2 < 0.5         ! get a point near Vbe = 0.5
  index = index + 1
  v2 = -ve[index]
END WHILE
i2 = ic.m[index]
! extract IS & NF
vt = 8.62e-5 * (TNOM + 273.15) ! thermal voltage
NF = 1 / vt * (v2 - v1) / log(i2 / i1)
IS = sqrt(i1 * i2) / exp((v1 + v2) / (2 * NF * vt))
print "IS = ";IS;"   NF = ";NF
print "... end of custom extraction ..."
```

Bipolar Device Model

The UCB bipolar transistor model is a hybrid of the Ebers-Moll [2] and Gummel-Poon [3] models. With a minimum parameter specification of IS, BF, and BR, the model defaults to the more simple Ebers-Moll model. The Ebers-Moll model is ideal because it neglects base width modulation, parasitic resistances, and high current injection effects.

Inclusion of additional parameters activates elements of the Gummel-Poon integral charge control model, which can provide greater accuracy. The Gummel-Poon model provides superior representation of the current flow in the transistor's base. It also provides accurate representation of parasitic resistances at all terminals, capacitance across all junctions, current-frequency effects, and temperature effects.

Simulators

The UCB bipolar model is supported by all SPICE simulators currently included with IC-CAP: HPSPICE, SPICE2(G6), and SPICE3.

NOTE

Simulators are provided as a courtesy to IC-CAP users; they are not supported by Agilent Technologies.

The default nominal temperature for HPSPICE is 25°C. For SPICE2 and SPICE3 it is 27°C. To force a nominal temperature, set the *TNOM* variable to the desired value.

Model Parameters

Model parameter extractions are based on the concept that, under steady-state conditions, specific sets of parameters uniquely simulate device performance. This allows extractions to be performed over isolated regions of the device's electrical response.

Forward and reverse DC bias extractions and junction capacitance characteristics are virtually independent of each other. Series resistance and small-signal high-frequency extractions depend on DC and capacitance parameters.

Model parameter extractions produce parameters that are referenced to a temperature of 27°C. To perform extractions at other temperatures, change the system variable *TEMP* to the correct value.

Table 1 provides definitions and SPICE default values of the bipolar model parameters, which fall into four primary categories: DC, capacitance, AC, and temperature effects. DC parameters are divided into three categories: DC forward, DC reverse, and series resistance. The parameter values are displayed in the Model Parameters folder.

Table 2 lists setup attributes.

Table 1 UCB Bipolar Transistor Parameters

Name	Description	Default
DC Large Signal Forward Bias		
BF	Ideal Maximum Forward Beta. Basic parameter for Ebers-Moll and Gummel-Poon models.	100
IKF	Knee Current for Forward Beta High Current Roll-off. Models variation in forward Beta at high collector currents. Use if device is to be used with high collector currents.	∞ Amp
IS	Transport Saturation Current. Basic parameter for Ebers-Moll and Gummel-Poon models.	1×10^{-16} Amp

Table 1 UCB Bipolar Transistor Parameters

Name	Description	Default
ISE	Base Emitter Leakage Saturation Current. Models variation in forward Beta at low base currents. Use if device is to be used with low base emitter voltage.	0 Amp
NE	Base Emitter Leakage Emission Coefficient. Models variation in forward Beta at low base currents. Use if device is to be used with low base emitter voltage.	1.5
NF	Forward Current Emission Coefficient. Used to model deviation of emitter base diode from ideal (usually approximately 1).	1.0
VAF	Forward Early Voltage. Models base collector bias effects. Used to model base collector bias on forward Beta and IS.	∞ volt
DC Large Signal Reverse Bias		
BR	Ideal Maximum Reverse Beta. The basic parameter for both Ebers-Moll and Gummel-Poon models. Use when the transistor is saturated or operating in reverse mode.	1.0
IKR	Knee Current for Reverse Beta High Current Roll-off. Specifies variation in reverse Beta at high emitter currents. Needed only if transistor is operated in reverse mode.	∞Amp
ISC	Base Collector Leakage Saturation Current. Specifies variation in reverse Beta at low base currents. Models base current at low base collector voltage. Use only if transistor is operated in reverse mode.	0 Amp
NC	Base Collector Leakage Emission Coefficient. Specifies variation in reverse Beta at low currents. Models base current at low base collector voltage. Use only if transistor is operated in reverse mode.	2.0
NR	Reverse Current Emission Coefficient. Used to model deviation of base collector diode from the ideal (usually about 1).	1.0
VAR	Reverse Early Voltage. Models emitter base bias effects. Use to model emitter base bias on reverse Beta and IS.	∞ Volt
Series Resistance		

Table 1 UCB Bipolar Transistor Parameters

Name	Description	Default
IRB	Base Resistance Roll-off Current. Models the base current at which the base resistance is halfway between minimum and maximum.	∞ Amp
RB	Zero Bias Base Resistance. Maximum value of parasitic resistance in base.	0 Ohm
RBM	Minimum Base Resistance. The minimum value of base resistance at high current levels. Models the way base resistance varies as base current varies.	RB Ohm
RC	Collector Resistance. Parasitic resistance in the collector. Important in high current and high frequency applications.	0 Ohm
RE	Emitter Resistance. Parasitic resistance in the emitter. Important in small signal applications.	0 Ohm
Capacitance		
CJC	Base Collector Zero Bias Capacitance. Helps model switching time and high frequency effects.	0 Farad
CJE	Base Emitter Zero Bias Capacitance. Helps model switching time and high frequency effects.	0 Farad
CJS	Zero Bias Substrate Capacitance. Helps model switching time and high frequency effects.	0 Farad
MJC	Base Collector Junction Grading Coefficient. Models the way junction capacitance varies with bias.	0.33
MJE	Base Emitter Junction Grading Coefficient. Models the way junction capacitance varies with bias.	0.33
MJS	Substrate Junction Grading Coefficient. Models the way junction capacitance varies with bias.	0.33
VJC	Base Collector Built-in Potential. Models the way junction capacitance varies with bias.	0.75 Volt
VJE	Base Emitter Built-in Potential. Models the way junction capacitance varies with bias.	0.75 Volt
VJS	Substrate Junction Built-in Potential. Models the way junction capacitance varies with bias.	0.75 Volt

Table 1 UCB Bipolar Transistor Parameters

Name	Description	Default
XCJC	Fraction of Base Collector. Capacitance that connects to the internal base node. Important in high frequency applications.	1.0
FC	Coefficient for Forward Bias Capacitance Formula. Provides continuity between capacitance equations for forward and reverse bias.	0.5
AC Small-Signal		
ITF	High Current Parameter for Effect on TF. Models decline of TF with high collector current.	∞ Amp
PTF	Excess Phase at FT. Models excess phase at FT.	0 Degree
TF	Ideal Forward Transit Time. Models finite bandwidth of device in forward mode.	0 Sec
TR	Ideal Reverse Transit Time. Models finite bandwidth of device in reverse mode.	0 Sec
VTF	Voltage Describing TF Dependence on Base-Collector Voltage. Models base-collector voltage bias effects on TF.	∞ Volt
XTF	Coefficient for Bias Dependence on TF. Models minimum value of TF at low collector-emitter voltage and high collector current.	0
Temperature Effects		
EG	Energy Gap for Modeling Temperature. Effect on IS, ISE, and ISC. Used to calculate the temperature variation of saturation currents in the collector, and base-emitter and collector base diodes.	1.11EV
XTB	Forward and Reverse Beta Temperature Exponent. Models the way Beta varies with temperature.	0
XTI	Temperature Exponent for Modeling. Temperature Variation of IS. Models the way saturation current varies with temperature.	3.0
TNOM	This global variable can be assigned temperature values in degrees C, for use by extractions and simulations.	27°C

Table 2 UCB Bipolar Model Setup Attributes

DUT / Setup	Inputs	Outputs	Transform	Function	Extractions
dc / fearly	vb, vc, ve, vs	ic	none	none	VAF (from rearly setup)
dc / rearly	vb, vc, ve, vs	ie	evextract	BJTDC_vaf_var	VAF, VAR
dc / fgummel	vb, vc, ve, vs	ib, ic	beta	equation: ic / ib	none
			isextract	BJTDC_is_nf	IS, NF
			fgextract	BJTDC_fwd_gummel	BF, IKF, ISE, NE
			optim1	Optimize	IS, NF
			optim2	Optimize	BF, IKF, ISE, NE
dc / rgummel	vb, vc, ve, vs	ib, ie	beta	equation: ie / ib	none
			nrextract	BJTDC_nr	NR
			rgextract	BJTDC_rev_gummel	BR, IKR, ISC, NC
			optimize	Optimize	BR, IKR, ISC, NC
cbe / cj	vbe	cbe	extract	Optimize	CJE, VJE, MJE
			cjfunc	PNCAPsimu	none: simulates c vs v
			set_CJ	Program	initial zero bias CJE
cbc / cj	vbc	cbc	extract	Optimize	CJC, VJC, MJC
			cjfunc	PNCAPsimu	none: simulates c vs v
			set_CJ	Program	initial zero bias CJC
ccs / cj	vcs	ccs	extract	Optimize	CJS, VJS, MJS
			cjfunc	PNCAPsimu	none: simulates c vs v
			set_CJ	Program	initial zero bias CJS
prdc / reflyback	ib, ic, ve, is	vc	extract	BJTDC_re	RE
prdc / rcsat	vb, vc, ve, vs	ic	extract	BJTDC_rc	RC (saturation)

Table 2 UCB Bipolar Model Setup Attributes (continued)

DUT / Setup	Inputs	Outputs	Transform	Function	Extractions
prdc / rcactive	vb, vc, ve, vs	ib,ic	RC_active	Program	RC (active)
prdc / rbbib	vb, vc, ve, vs	ib	rbb	RBBcalc	none: calc rb vs ib
ac / rbbac	vb, vc, ve, vs, freq	h	extract	BJTAC_rb_rbm_irb	RB, RBM, IRB
			h11corr	H11corr	corrects H11 for Zout
			htos	TwoPort	none: h-par to s-par
ac / h21vsvbe	vb, vc, ve, vs, freq	h	acextract	BJTAC_high_freq	TF, ITF, XTF, VTF, PTF
			scale_params	Program	none: scales AC parameters
ac / h21vsvbc	vb, vc, ve, vs, freq	h	extract_TR	Optimize	TR

Test Instruments

The HP 4141, HP/Agilent 4142, or HP 4145 can be used to derive DC model parameters from measured DC voltage and current characteristics.

The HP 4271, HP 4275, HP 4280, HP/Agilent 4284, or HP 4194 can be used to derive capacitance model parameters from measured capacitance characteristics at the device junctions.

Instrument-to-Device Connections

When the device is installed in a test fixture, verify the correct connection of device nodes by checking the inputs and outputs for the appropriate DUTs. The following table is a cross- reference of the connections between the terminals of a typical bipolar transistor and various measurement units. These connections and measurement units are defined in the model file.

Table 3 Instrument-to-Device Connections

DUT	Collector	Base	Emitter	Substrate	Comments
dc	SMU1	SMU2	SMU3	SMU4	
cbe	open	CM(H)	CM(L)	open	calibrate for stray capacitance
cbc	CM(L)	CM(H)	open	open	calibrate for stray capacitance
ccs	CM(H)	open	open	CM(L)	calibrate for stray capacitance
prdc	SMU1	SMU2	SMU3	SMU4	
ac	NWA (Port2) and SMU1	NWA (Port1) and SMU2	ground	SMU4	calibrate for reference plane

Notes:
1. DUT is the name of the DUT as specified in DUT-Setup.
2. To read the table: *dc* has the dc measurement unit SMU1 connected to its collector, SMU2 connected to its base, SMU3 connected to its emitter, and SMU4 connected to its substrate.

Input and output tables in the various setups use abbreviations C (collector), B (base), E (emitter), and S (substrate) for the bipolar transistor nodes. These nodes are defined in the Circuit folder.

Measurement units (abbreviated as follows) are defined in Hardware Setup.

SMU# for DC measurement units

VM# for voltage monitor units

VS# for voltage source units

CM for capacitance measurement units

NWA for network analyzer port units

Measuring and Extracting

Bipolar parameter extraction is divided into four categories: DC, capacitance, parasitic resistance, and AC. These categories correspond to the required supporting measurements described under Test Instruments.

The *bjt_npn.mdl* file provides DUTs and setups that correctly bias a typical device for the measurements needed to perform the associated parameter extractions.

Measurement and Extraction Guidelines

The following guidelines are provided to help you achieve more successful model measurements and extractions.

Setting Instrument Options

Before starting a measurement, you can quickly verify the instrument options settings. Save the current instrument option settings by saving the model file to *<file_name>.mdl*. Some of the Instrument Options specify instrument calibration. For the most accurate results, calibrate the instruments before taking IC-CAP measurements.

Typical DC and cv instrument options are:

- DC measurements are generally taken with Integration Time = Medium.
- CV measurements in the *femtofarad* region usually require High Resolution = Yes and Measurement Freq (kHz) = 1000.

When taking AC measurements with a network analyzer, several instrument settings are critical. And, calibration must be performed on structures that have impedances similar to the stray parasitics of the device under test (DUT).

Typical AC instrument options are:

- Input power to the device is typically –10 to –30dBm (after port attenuation).
- Setting the averaging factor in the 2 to 4 range reduces measurement noise.

- Because IC-CAP requires the instrument to perform error correction, set Use Internal Calibration = Yes.

NOTE

The error terms saved to file during a network analyzer software calibration are not identified by error code.

The order shown below represents the order in which they are saved and displayed in IC-CAP:

0. EDF [directivity]
1. EDR [directivity]
2. EXF [isolation]
3. EXR [isolation]
4. ESF [source match]
5. ERF [ref freq response]
6. ESR [source match]
7. ERR [ref freq response]
8. ELF [load match]
9. ETF [trans freq response]
10. ELR [load match]
11. ETR [trans freq response]

Experiment with the other network analyzer options to obtain the best results with specific devices.

Measuring Instruments

Ensure that the measuring instruments (specified by unit names in the inputs and outputs) are correctly connected to the DUT. Refer to Table 3 for a list of nodes and corresponding measurement units. The quality of the measuring equipment (instruments, cables, test fixture, transistor sockets, and probes) can influence the noise level in the measurements.

The series resistance in test fixtures can also be critical when making high current measurements. For example, a 1-ohm resistance in series with the emitter at an $Ic = 20\ mA$ can cause a factor-of-two error in the measured versus simulated DC performance. Series resistances that are not accounted for in the device model can be included by adding them to the test circuit for the DUT.

Ensure that all characteristics of the measurement stimulus and corresponding measured response are specified in the respective input and output tables.

For some measurements, the instruments or test hardware must be calibrated to remove non-device parasitics from the DUT. For bipolar devices, stray capacitance due to probe systems, bond pads, and so on should be calibrated out prior to each measurement.

In making high-frequency two-port measurements with a network analyzer, the reference plane of the instrument must be calibrated out to the DUT. IC-CAP relies on the internal calibration of the instruments for full error-corrected data. Calibration using OPEN, SHORT, THRU, and 50-ohm LOADS must be correctly performed.

Extracting Model Parameters

For a given setup, you can find the extraction transforms in the Extract/Optimize folder. IC-CAP's extraction algorithms exist as functions; choose **Browse** to list the functions available for a setup.

When the *Extract* command is selected from the setup, all extractions in the setup are performed in the order listed in the setup. This order is usually critical to proper extraction performance. Extractions are typically completed instantly and the newly extracted model parameter values are placed in Model Parameters.

Simulating Device Response

Simulation uses model parameter values currently in Model Parameters. A SPICE deck is created and the simulation performed. The output of the SPICE simulation is then read into IC-CAP as simulated data.

Select a simulator from Tools > Hardware Setup or define a *SIMULATOR* variable. Simulations vary in the amount of time they take to complete. DC simulations generally run much faster than cv and AC simulations.

If simulated results are not as expected, use the Simulation Debugger (Tools menu) to examine the input and output simulation files. The output of manual simulations is not available for further processing by IC-CAP functions (such as transforms and plots). For more information refer to "Using the Simulation Debugger" in the *IC-CAP User's Guide*.

Displaying Plots

The Display Plot function displays all graphical plots defined in a setup. The currently active graphs are listed in the Plots folder in each setup.

Measured data is displayed as a solid line; simulated data is displayed as a dashed or dotted line of the same color. After an extraction and subsequent simulation, view the plots for agreement between measured and simulated data. Plots are automatically updated each time a measurement or simulation is performed.

Optimizing Model Parameters

Optimization of model parameters improves the agreement between measured and simulated data. The bipolar model typically requires very little optimization because most of the extraction algorithms have some optimization built into them.

Capacitance parameter extractions are actually done through optimization. An Optimize Transform whose *Extract Flag* is set to *Yes* is automatically called after any extraction that precedes it in the Transform list.

Optimizing AC parameters can be very time-consuming because of the number of SPICE simulations required.

PNP Transistors

In the *bjt_pnp.mdl* file included with IC-CAP, PNP transistors are measured, extracted, and simulated in a manner similar to NPN transistors. The critical difference with a PNP device is that the bias voltages are of opposite polarity from an NPN device.

To extract the two models using the same algorithms, set a variable in Model Variables; *POLARITY* should have the value *PNP*. The extraction default NPN will result in incorrect parameter values or extraction errors on PNP data.

Another variable convenient for displaying PNP Plots is *inv_plot*. This variable can invert the plots in *bjt_pnp.mdl* so they plot in the same direction as NPN plots. Set *inv_plot* to -1 to do this.

Extracting Parameters

This section describes the general process for extracting model parameter data from the UCB bipolar transistor. The process applies to all types of parameters: DC, capacitance, and AC. The differences between extracting one type of parameter and another are primarily in the types of instruments used to measure the data and the specifications within the DUTs and setups.

Parameters are typically extracted from measured data but can also be extracted from simulated data. To extract from measured data, ensure that the outputs specified in the extraction transforms use the *.m* suffix. For example, IS and NF are extracted using the *BJTDC_is_nf* function. To extract from measured data, IC-CAP uses *log10(ic.m)* as the specification of the forward collector current. (Use the *.s* suffix when extracting from simulated data.)

When performing an extraction, accurate results depend on the sequence of steps. The top-to-bottom order of DUTs and setups in a model file is the suggested order of measurements and extractions. In the *bjt_npn.mdl* file, the large signal DC and junction capacitance parameters are independent of each other. However, for the parasitic resistances and AC parameters to be accurately extracted, the preceding two groups must be successfully extracted first. Setups in *bjt_npn.mdl* are designed for use with a typical bipolar transistor. You may be able to improve results with your own devices by modifying these setups to more closely conform to your needs.

The general extraction procedure is summarized next, starting with the measurement process.

1 Install the device to test in a test fixture and connect the measuring instruments.

2 Ensure the test fixture, signal sources and measuring instruments, and workstation are physically and logically configured to the IC-CAP system.

3 Load the model.

4 Select the DUT and setup.

NOTE Execute measurements and extractions in the order listed in *DUTs-Setups* to ensure the correct order; otherwise, incorrect results may result.

5 Issue the *Measure* command.

6 Issue the *Extract* command.

7 Issue the *Simulate* command.

8 Display the results.

9 Fine tune the extracted parameters if needed by optimizing.

DC Large-Signal Parameters

Setups are provided for measuring and extracting the properties of the internal transistor (not including parasitic resistances); these are *fearly, rearly, fgummel,* and *rgummel.*

The *fearly* and *rearly* setups measure forward and reverse Early voltage characteristics, respectively. The Early voltage parameters VAF and VAR are extracted simultaneously in the *rearly* setup, using measurements taken from both *fearly* and *rearly.*

The *fgummel* and *rgummel* setups perform the forward and reverse Gummel plot measurements. Parameters IS, BF, NE, IKF, ISE, and NF are extracted by the *fgummel* setup, while the extractions in the *rgummel* setup produce the BR, NR, IKR, ISC, and NC parameters. The model uses the saturated current parameter IS to simulate current flow in both directions.

Junction Capacitance Parameters

Measuring bipolar transistor capacitance characteristics requires three DUTs. This is because each p-n junction is a physically different one-port connection. The base-emitter, base-collector, and collector-substrate junctions each have a different DUT and setup. While you will perform all three measurements on the same physical device, each measurement requires different instrument connections for the corresponding DUT and setup. The DUTs and instrument connections for each measurement are listed in Table 3.

Each p-n junction is measured from a small forward bias to a large reverse bias. The extractions are performed using the transform *set_CJ* to find the initial value of CJ0, then optimizing the parameters of the general p-n junction capacitance equation to the measured data. This produces the capacitance, built-in voltage, and grading factors for each DUT: CJE, VJE, MJE; CJC, VJC, MJC; CJS, VJS, MJS. For the most accurate extractions, calibrate out stray capacitance from cables, probes, and bond pads before taking each p-n junction capacitance measurement.

DC Parasitic Resistance Parameters

Three parasitic resistances are connected to the bipolar transistor: RE, RC, and RB. RE and RC are constant value components, while RB is a function of base current. RE is measured by the setup *reflyback*. This setup saturates the transistor, then measures the differential voltage drop from collector to emitter (with Ic = 0) versus the differential base-to-emitter current.

RC is measured by the setup *rcsat,* which measures the parameter as the DC resistance from collector to emitter at the onset of saturation. Alternately, the *rcactive* setup can be used to measure the collector resistance in the active region of device operation. However, this extraction is dependent on the operating point, which must be specified by manually placing a box on the Plot contained in the setup. For complete information on using this extraction, refer to HP Application Note *Advanced Bipolar Transistor Modeling Techniques*[1].

The setup *rbbib* does not actually measure or extract RB. Instead, it produces a characteristic curve of base-to-emitter bias versus DC base current. The resulting curve is used when the base resistance is measured and extracted using S-parameters.

Base Resistance and Transit Time Parameters

The AC DUT uses setups that measure S-parameters with a network analyzer. The quality of the measured S-parameters depends on the calibration of the network analyzer. IC-CAP does not perform error correction; it relies totally on the measuring instruments for the correction of errors.

Making high-frequency measurements on packaged transistors can lead to unexpected results. This is because of the stray capacitance and inductance that are a part of the package. Measure S-parameters with a high-quality microwave wafer probe.

The AC setup *rbbac* measures H11 of the transistor in the common emitter mode. This input impedance is then used in the extraction to produce model parameters RB, IRB, and RBM.

The AC setup *h21vsvbe* measures H21 of the transistor in the common emitter mode. The measured current gain is then used to extract a small signal model that produces the parameters TF, ITF, VTF, XTF, and PTF.

The AC setup *h21vsvbc* measures H21 of the transistor in the common collector mode. The measured current gain is then used to extract the parameter TR.

Extraction Algorithms

This section describes the extraction algorithms used for DC, capacitance, parasitic resistance, and AC model parameters of the bipolar transistor.

DC Parameter Extractions

The Early voltage extractions produce the model parameters VAF and VAR. The output conductance of *Ic* versus *Vce* for steps of *Vb* is used in the calculation. Both Early parameters are extracted simultaneously, which requires both forward and reverse measurements prior to extraction. The actual extraction is performed under the *rearly* setup. The substrate bias should be held at a negative (positive) voltage for an NPN (PNP) transistor. For the extraction to function correctly, the device must be completely out of saturation at the 20 percent point of each curve, and the forward and reverse curves must have the same number of steps.

The forward Gummel measurement is used to extract IS, NF, BF, IKF, ISE, and NE. This measurement holds the base-collector voltage at approximately 0V and drives the emitter with a negative bias sweep. The bias should produce *Ic* in the range of less than 1nA to more than 10 mA for a typical IC transistor.

First, IS and NF are extracted from the low current region of the *Ic* versus *Vbe* data using a least-squares fit. The very low current region of the *Ib* versus *Vbe* data is used to obtain ISE and NE, the base recombination parameters. An internal optimization in the extraction algorithm is then used to produce BF and IKF and fine-tune the ISE and NE parameters. If insufficient high current data is available, IKF will be set to a default value of 10A. To guarantee that IKF is extracted, measure until beta has rolled off to approximately half of its peak value.

The reverse Gummel measurement is used to extract NR, BR, IKR, ISC, and NC. This measurement and extraction is analogous to the forward measurement except that the transistor is now in the reverse active mode. If the measurement is made on an IC structure that has a substrate of opposite

polarity to the collector, it is possible that the plot of reverse *Beta* versus *Ie* will not fit well. This is because the parasitic transistor formed by the base- to- collector- to- substrate begins to conduct, thus robbing current from the base of the transistor being modeled. There is a solution to this problem. The example file *npnwpnp.mdl* includes a compound structure of both an NPN transistor and its parasitic PNP device. This model allows you to produce an excellent fit of the NPN transistor operating in the reverse bias region. Refer to "Circuit Parameter Extraction" on page 545 for more information on using this file to characterize the reverse active mode of operation.

Capacitance Parameter Extractions

The capacitances are split into three different DUTs. The measurement is performed over a range of small forward bias (where v < VJ · FC) to at least several volts of reverse bias. The parameter extraction is accomplished through optimization of the controlling parameters in the characteristic equation for the junction capacitance. The extraction from each produces the zero bias capacitance CJx, the built-in potential of the junction VJx, and the grading factor of the junction MJx. The forward bias coefficient FC is set to the SPICE default value of 0.5. The purpose of this parameter is to switch the capacitance in the simulator into a linear model before the junction bias approaches VJx.

Parasitic DC Parameter Extractions

This set of setups uses DC measurements to obtain the emitter resistance RE, the collector resistance RC, and a DC I versus V relationship to be used later in the base resistance extraction. RE is extracted from a measurement of the differential of collector voltage with respect to base current with the transistor biased into saturation. A linear fit is performed on the part of the curve that is most sensitive to the effects of RE.

In the *rcsat* setup RC is extracted from a measurement of *Ic* versus *Vce* with the base biased so that the transistor is near its peak Beta point and well into saturation. The extraction uses a

linear fit along with the known RE. In the *rcactive* setup RC is extracted at a bias selected by placing a box on the Plot of *Ic* versus *Vce*.

The setup *rbbib* is not actually used to extract any model parameters directly but is used by the following AC measurements in the extraction of the base resistance parameters. The base voltage bias specification used in this setup and in the *rbbac* setup must be the same. To facilitate this, the start value, stop value, and number of points are set using four variables in the model level variable table. These are *rbbstart, rbbstop, rbbnpts,* and *rbbvc*. The start and stop bias voltages should sweep the transistor's operating point from near peak Beta to well into Beta roll-off.

AC Parameter Extractions

Base resistance and transit time parameters are extracted from network analyzer measurements of the transistor's S-parameters converted to H-parameters. Both of these sets of model parameters are highly dependent upon the prior extraction of the DC, capacitance, and parasitic resistance parameters.

The base resistance is extracted from H11 data versus frequency and bias. The H11 data traces a circular path on a Re-Im axis system versus frequency. The measurement frequency should be held low enough so that this circular pattern does not start to become linear. This characteristic is used to obtain the real value of Rbase versus base current. From the characteristic of Rbase versus Ibase, the RB, IRB, and RBM parameters are extracted.

The transit time parameters, TF, XTF, ITF, VTF, and PTF, are extracted from measurements of the common-emitter current gain H21. The measurement frequency should be higher than the -3dB roll-off frequency of the transistor at all bias levels. However, the measurement frequency should also be low enough so that the magnitude of H21 over the bias levels is always greater than 2.0. Regions of the H21 versus *Vbe* versus *Vce* data are isolated where each of these parameters has a dominating effect on an extraction performed there. The

extractions use an optimization routine that matches the performance of the complete small signal model to the measured data. The extraction assumes that all other model parameters have been accurately obtained. If the H21 measurement has not calibrated out the stray capacitance (from bond pads, package, probe, or others) the initial extraction may fail and an extraction decoupled from the small signal model will be performed. These resulting parameters may need scaling using the *scale_params* transform, depending on any unaccounted stray capacitance in the small signal model.

The reverse transit time parameter TR is extracted from measurements of the common-collector current gain H21.

IC-CAP supports two different methods of calculating the Q1 component of the base charge during the extractions.

$$Q1 = \frac{1}{1 - \frac{Vbe}{VAR} - \frac{Vbc}{VAF}} \quad \text{(default method)}$$

$$Q1 = 1 + \frac{Vbe}{VAR} + \frac{Vbc}{VAF} \quad \text{(alternate method)}$$

The alternate method can be selected by defining a model parameter or variable named GPQ1 and setting it equal to 0. If GPQ1 is not defined or non-zero, the default method is used.

References

1 *Advanced Bipolar Transistor Modeling Techniques,* HP Application Note 1201-4, Publication #5091-2503EUS, July, 1991.

2 J.J. Ebers and J.L. Moll, *Large-Signal Behaviour of Junction Transistors,* Proc. IRE, No. 42, 1954.

3 H.K. Gummel and H.C. Poon, *An Integral Charge Control Model of Bipolar Transistors,* Bell Systems Technical Journal, No. 49, 1970.

2 MOSFET Characterization

UCB MOSFET Model 41
Simulators 42
MOSFET Model Parameters 43
Test Instruments 49
Measuring and Extracting 51
Extraction Algorithms 61
HSPICE LEVEL 6 MOSFET Model 64
References 68

This chapter describes the UC Berkeley MOSFET transistor model supported in SPICE. Descriptions of model setup, instrument connections, and model parameters are included as well as test instrument information. Information is included for making DC and capacitance measurements and their corresponding extractions.

NOTE The HSPICE LEVEL 6 MOSFET model is an enhanced version of the MOSFET LEVEL 2 model; refer to the section "HSPICE LEVEL 6 MOSFET Model" on page 64 for parameter measurement and extraction information.

The IC- CAP MOSFET modeling module provides setups that can be used for general measurement and model extraction for MOS devices. Four example files are provided for the MOSFET model; the files can also be used as a template for creating custom model configurations.

nmos2.mdl extracts parameters for the LEVEL 2 N-channel model

pmos2.mdl extracts parameters for the LEVEL 2 P-channel model

nmos3.mdl extracts parameters for the LEVEL 3 N-channel model

pmos3.mdl extracts parameters for the LEVEL 3 P-channel model

The IC-CAP system offers the flexibility to modify any measurement or simulation specification.

The model extractions provided are also intended for general MOS IC processes. If you have another method of extracting specific model parameters, you can do so with the Program function or by writing a function in C and linking it to the function list. For Program function details or for writing user-defined C-language routines, refer to Chapter 9, "Using Transforms and Functions," in the *IC-CAP User's Guide*.

UCB MOSFET Model

The UCB MOSFET model is fully compatible with the UCB model developed for use with the UCB SPICE simulator. The model is actually a combination of three models, each being specified by an appropriate value of the LEVEL parameter. After specifying the model, enter the correct set of parameters for that model. Some of these parameters are shared between different models, while others affect only a specific model.

Extraction for the LEVEL 1 model (Shichman-Hodges) is not supplied with this release of IC-CAP. The LEVEL 2 model [1] is an advanced version of LEVEL 1, and can use either electrical or process type parameters. The LEVEL 3 [1] model is semi-empirical because it uses parameters that are defined by curve fitting rather than by device physics.

Simulators

The MOSFET model is supported by the SPICE simulators included with IC-CAP: SPICE2, SPICE3 and HPSPICE. The model files provided can also be used with the HSPICE simulator and, with some modification, the Saber simulator.

NOTE Simulators are provided as a courtesy to IC-CAP users; they are not supported by Agilent Technologies.

The default nominal temperature for HPSPICE is 25°C; for SPICE2 and SPICE3 it is 27°C. To force a nominal temperature, set the *TNOM* variable to the desired value.

MOSFET Model Parameters

Table 4 lists parameters for the three model levels according to DC and cv extraction in IC-CAP. (Some of these parameters are redundant and therefore only a subset of them is extracted in IC-CAP.) Table 5 describes model parameters by related categories and provide default values. The parameter values are displayed in the Circuit folder. Table 6 lists setup attributes.

Table 4 Summary of UCB MOSFET Controlling Model Parameters

Type	LEVEL 1	LEVEL 2†	LEVEL 3†
Classical	VTO, GAMMA, PHI KP IS, JS, TOX	NSUB, UO, UCRIT, UEXP, UTRA, NFS, NSS, TPG	NSUB, UO, THETA, NFS, NSS, TPG
Short-channel		LD, XJ	LD, XJ
Narrow-width			DELTA, WD†††
Saturation	LAMBDA	NEFF, VMAX	ETA, KAPPA
External resistance	NRD††, NRS†† RD, RS		
Junction capacitance	AD††, AS††, CBD, CBS CJ, FC, MJ, PB		
Sidewall capacitance	PD††, PS††, CSJW, MJSW		
Overlap capacitance	CGBO, CGDO, CGSO		
General	LEVEL, L††, W††		
IC-CAP Temperature Specification	TNOM (system variable)		

Notes:
† LEVEL 2 and LEVEL 3 also include LEVEL 1 parameters.
†† Indicates device parameters (model and device parameters are listed together).
††† WD does not exist in the SPICE UCB version; it has been added to some SPICE versions and is included in IC-CAP. If WD is not in your simulator, ignore the result (set to zero), or subtract 2·WD from the channel width. In the MOS model files provided with IC-CAP, the width specification W in each of the DUTs has been modified to subtract the value of 2·WD from the drawn width. WD is specified in Model Variables.

Table 5 UCB MOSFET Parameters

Name	Description	Default
Capacitance		
CGBO	Gate to Bulk Overlap Capacitance. Capacitance due to design rules that require the gate be extended beyond the channel by some amount. Not voltage dependent. Total Cgb capacitance equals Cgbo times channel length.	0 F/m
CGDO	Gate to Drain Overlap Capacitance. Capacitance due to the lateral diffusion of the drain in an Si gate MOSFET. Not voltage dependent. Total Gcd capacitance equals Cgdo times the channel width.	0 F/m
CGSO	Gate to Source Overlap Capacitance. Capacitance due to the lateral diffusion of the source in an Si gate MOSFET. Not voltage dependent because it is not a junction capacitance. Total Cgs capacitance equals Cgso times channel width.	0 F/m
CJSW	Zero Bias Junction Sidewall Capacitance. Models the nonlinear junction capacitance between the drain and the source junction sidewall. (Pd + Ps) * CJSW = total junction sidewall capacitance.	0 F/m
MJSW	Grading Coefficient of Junction Sidewall. Models the grading coefficient for the junction sidewall capacitance.	0.33
PB	Bulk Junction Potential. Models the built-in potential of the bulk-drain or bulk-source junctions. The default is usually adequate.	0.8 volt
FC	Forward Bias Non-Ideal Junction Capacitance Coefficient. Models the point (FC * PB) at which junction capacitance makes the transition between forward and reverse bias.	0.5
Electrical Process		
IS	Substrate Junction Saturation Current. Helps model current flow through the bulk-source or bulk-drain junction.	1×10^{-16} Amp
JS	Substrate Junction Saturation Current/m^2. Js equals Is divided by the junction area. For example, Isd=Js*Ad where Ad is the drain area.	1×10^{-4} A/m^2
RD	Drain Ohmic Resistance. This parameter is geometry independent in SPICE and IC-CAP. In fact, it is inversely proportional to channel width.	0 Ohm
UCRIT	Critical Field for Mobility Degradation. Used in level=2 model only.	1000 V · cm^{-1}
UEXP	Critical Field Exponent. Used in level=2 model only.	0
UO	Surface Mobility at Low Gate Levels. Specifies mobility in level=2 and level=3 models. In the level=2 model, if Kp is UTRA	600 cm^2/(V · S)

Table 5 UCB MOSFET Parameters (continued)

Name	Description	Default
UTRA	Transverse Field Coefficient. Used in level=2 model only. Set UTRA to 0 to obtain same result as SPICE.	0
VMAX	Maximum Drift Velocity of Carriers. Determines whether Vdsat is a function of scattering velocity limited carriers or a function of drain depletion region pinch-off.VMAX is valid only for level=2 and level=3 models. If VMAX is specified, the scattering velocity limited carrier model is used to determine Vdsat.	0 $m \cdot s^{-1}$
NEFF	Total Channel Charge. A multiplicative factor of NSUB, NEFF determines saturated output conductance. Used only in the level=2 model, and only when Vmax is specified.	1.0
Physical Process		
LD	Lateral Diffusion Coefficient. Used to determine the effective channel length.	0 Meter
TOX	Oxide Thickness. Used when calculating conduction factor, backgate bias effects, and gate-channel capacitances.	100×10^{-9} Meter
TPG	Type of Gate. Indicates whether gate is of metal or poly-silicon material (0=aluminum; 1=opposite substrate; -1=same as substrate). Used in calculating threshold voltage when Vto is not specified.	1
WD	Channel Width Reduction. Used to determine the effective channel width This parameter is assumed to be 0 in SPICE.	0 Meter
XJ	Metallurgical Junction Depth. Defines the distance into the diffused region around the drain or source at which the dopant concentration becomes negligible. Used to model some short channel effects.	0 Meter
Threshold Related		
NFS	Effective Fast Surface State Density. Used to determine subthreshold current flow. Not valid for extracting simple linear region classical parameters.	0 cm^{-2}
NSS	Effective Surface Charge Density. Used to calculate threshold voltage when Vto is not specified.	0 cm^{-2}
NSUB	Substrate Doping Concentration. Used in most calculations for electrical parameters. It is more accurate to specify Vto rather than deriving it from NSUB. However, NSUB should be specified when modeling the back gate bias dependency of Vto.	1×10^{15} cm^{-3}
DELTA	Width Effect on Threshold Voltage. Used in LEVEL=2 and LEVEL=3 models to shift threshold voltage for different channel widths.	0
ETA	Static Feedback. Used in LEVEL=3 model to decrease threshold for higher drain voltage.	0

Table 5 UCB MOSFET Parameters (continued)

Name	Description	Default
GAMMA	Bulk Threshold. The proportionality factor that defines the threshold voltage to backgate bias relationship. Used in the derivation of Vto, Ids, and Vdsat. If not specified in LEVEL=2 and LEVEL=3 models, it is computed from NSUB.	0 $V^{1/2}$
VTO	Extrapolated Zero Bias. Threshold Voltage Models the onset of strong inversion in the LEVEL=1 model. Marks the point at which the device starts conducting if weak inversion current is ignored.	0 Volt
Electrical		
KAPPA	Saturation Field Factor. Used in the level=3 model to control saturation output conductance.	0.2
KP	Intrinsic Transconductance. If not specified for the level=2 model, KP is computed from Kp=u0*Cox. In some of the literature, KP may be shown as k'. The default for the LEVEL=1 model is 2x10e-5.	0 A/V^2
LAMBDA	Channel Length Modulation Models. The finite output conductance of a MOSFET in saturation. It is equivalent to the inverse of Early Voltage in a bipolar transistor. Specifying this parameter ensures that a MOSFET will have a finite output conductance when saturated. In the level=1 model, if lambda is not specified a zero output conductance is assumed. In the level=2 model, if lambda is not specified, it will be computed.	0 V^{-1}
PHI	Surface Potential Models. The surface potential at strong inversion.If not specified in level=2 and level=3 models, it is computed as PHI=2kT/q *ln(Nsub/ni). PHI also may be shown as 2*PHIb.	0 Volt
THETA	Mobility Reduction. Used in level=3 to model the degradation of mobility due to the normal field.	0 V^{-1}
Device Geometry		
L	Drawn or Mask Channel Length. Physical length of the channel.	1×10^{-4} Meter
W	Drawn or Mask Channel Width. Physical width of channel.	1×10^{-4} Meter
AD	Area of Drain Area of drain diffusion. Used in computing Is (from Js), and drain and source capacitance from Cbd=CjAd.	0 m^2
AS	Area of Source diffusion. Can be used as described for AD.	0 m^2
NRD	Equivalent Squares in Drain Diffusion. Number of equivalent squares in the drain diffusion. Multiplied by Rsh to obtain parasitic drain resistance (Rd).	1.0
NRS	Equivalent Squares in Source Diffusion. Number of equivalent squares in the source diffusion. Multiplied by Rsh to obtain parasitic source resistance (Rs).	1.0

Table 5 UCB MOSFET Parameters (continued)

Name	Description	Default
PD	Drain Junction Perimeter. Used with CJSW and MJSW to model the junction sidewall capacitance of the drain.	0 Meter
PS	Source Junction Perimeter. Used with CJSW and MJSW to model the junction sidewall capacitance of the source.	0 Meter
General		
LEVEL	Extraction Level. Specifies one of four extraction levels.	1

Table 6 MOSFET Setup Attributes

DUT / Setup	Inputs	Outputs	Transform	Function	Extractions
LEVEL 2 Model					
large / idvg	vg, vb, vd, vs	id	extract	MOSDC_lev2_lin_large	NSUB, UO, UEXP, VTO
			optimize		NSUB, UO, UEXP, VTO
narrow / idvg	/ /	/ /	extract	MOSDC_lev2_lin_narrow	DELTA, WD
			optimize		DELTA, WD
short / idvg	/ /	/ /	extract	MOSDC_lev2_lin_short	LD, XJ
			optimize		LD, RD, RS, XJ
short / idvd	vd, vg, vb, vs	id	extract	MOSDC_lev2_sat_short	NEFF, VMAX
			optimize		NEFF, VMAX
cbd1 / cjdarea	vb, vd	cbd	set_CJ extract	Program	initial zero bias CJ
				Optimize	CJ, MJ, PB
cbd2 / cjdp3erimeter	vb, vd	cbd	extract	MOSCV_total_cap	CJ, MJ, CJSW, MJSW, PB
LEVEL 3 Model					
large / idvg	vg, vb, vd, vs	id	extract	MOSDC_lev3_lin_large	NSUB, UO, THETA, VTO
			optimize		NSUB, UO, THETA, VTO

Table 6 MOSFET Setup Attributes

DUT / Setup	Inputs	Outputs	Transform	Function	Extractions
narrow / idvg	/ /	/ /	extract	MOSDC_lev3_lin_narrow	DELTA, WD
			optimize		DELTA, WD
short / idvg	/ /	/ /	extract	MOSDC_lev3_lin_short	LD, RD, RS, XJ
			optimize		LD, RD, RS, XJ
short / idvd	vd, vg, vb, vs	id	extract	MOSDC_lev3_sat_short	ETA, KAPPA
			optimize		ETA, KAPPA
cbd1 / cjdarea	vb, vd	cbd	set_CJ extract	Program	initial zero bias CJ
				Optimize	CJ, MJ, PB
cbd2 / cjdperimeter	vb, vd	cbd	extract	MOSCV_total_cap	CJ, MJ, CJSW, MJSW, PB

Test Instruments

The HP 4141, HP/Agilent 4142, or HP 4145 can be used to derive DC model parameters from measured DC voltage and current characteristics.The HP 4271, HP 4275, HP 4280, HP/Agilent 4284, or HP 4194 can be used to derive capacitance model parameters from measured capacitance characteristics at the device junctions.

Instrument-to-Device Connections

When the device is installed in a test fixture, verify the correct connection of device nodes by checking the inputs and outputs for the DUTs. Table 7 is a cross-reference of connections between the terminals of a typical MOSFET device and various measurement units. These connections and measurement units are defined in the model file.

Input and output tables in the various setups use abbreviations D (drain), G (gate), S (source), and B (bulk [substrate]) for the MOSFET device nodes. These nodes are defined in the Circuit folder.

Measurement units (abbreviated as follows) are defined in Hardware Setup.

SMU# for DC measurement units
VS# for voltage source units
VM# for voltage monitor units
CM for capacitance measurement units
NWA for network analyzer ports units

Table 7 Instrument-to-Device Connections

DUT	Drain	Gate	Source	Bulk	Comments
large	SMU1	SMU2	SMU3	SMU4	
narrow	SMU1	SMU2	SMU3	SMU4	
short	SMU1	SMU2	SMU3	SMU4	

Table 7 Instrument-to-Device Connections (continued)

DUT	Drain	Gate	Source	Bulk	Comments
cbd1	CM(L)	open	open	CM(H)	calibrate for parasitic capacitance
cbd2	CM(L)	open	open	CM(H)	calibrate for parasitic capacitance

Notes:
DUT is the name of the DUT as specified in DUT-Setup. Example: DUT *large* has the DC measurement unit SMU1 connected to its drain, SMU2 connected to its gate, SMU3 connected to its source, and SMU4 connected to its bulk

Measuring and Extracting

This section provides guidelines as well as procedures for performing measurements and extractions of MOSFET devices.

Measurement and Extraction Guidelines

The following guidelines are provided to help you achieve more successful model measurements and extractions.

Setting Instrument Options

Before starting a measurement, you can quickly verify instrument options settings. Save the current instrument option settings by saving the model file to *<file_name>.mdl* from the model window. Some of the Instrument Options specify instrument calibration. For the most accurate results, calibrate the instruments before taking IC-CAP measurements.

Typical DC and cv instrument options are:

- DC measurements are generally taken with *Integration Time = Medium*.
- CV measurements in the femtofarad region usually require High *Resolution = Yes* and *Measurement Freq (kHz) = 1000*.

Measuring Instruments

Ensure that the measuring instruments (specified by unit names in the inputs and outputs) are correctly connected to the DUT. Refer to Table 7 for a list of nodes and corresponding measurement units. The quality of the measuring equipment (instruments, cables, test fixture, transistor sockets, and probes) can influence the noise level in the measurements and extracted parameter values.

For some measurements the instruments or test hardware must be calibrated to remove non-device parasitics from the DUT. For MOS devices, stray capacitance due to probe systems, bond pads, and so on should be calibrated out prior to each measurement.

Extracting Model Parameters

For a given setup, you can find the extraction transforms in the Extract/Optimize folder. IC-CAP's extraction algorithms exist as functions; choose **Browse** to list the functions available for a setup.

When the *Extract* command is selected from the setup, all extractions in the setup are performed in the order listed in the setup. This order is usually critical to proper extraction performance. Extractions are typically completed instantly and the newly extracted model parameter values are placed in Model Parameters.

Simulating Device Response

Simulation uses model parameter values currently in Model Parameters. A SPICE deck is created and the simulation performed. The output of the SPICE simulation is then read into IC-CAP as simulated data.

Select a simulator from Tools > Hardware Setup or define a *SIMULATOR* variable. DC simulations generally run much faster than cv simulations. CV simulations can be done in a much shorter time by executing the *calc_mos_cbd_model* transform instead of running the simulator.

If simulated results are not as expected, use the Simulation Debugger (Tools menu) to examine the input and output simulation files. The output of manual simulations is not available for further processing by IC-CAP functions (such as transforms and plots). For more information refer to "Using the Simulation Debugger" in the *IC-CAP User's Guide*.

Displaying Plots

The Display Plot function displays all graphical plots defined in a setup. The currently active graphs are listed in the Plots folder in each setup.

Measured data is displayed as a solid line; simulated data is displayed as a dashed or dotted line. After an extraction and subsequent simulation, view the plots for agreement between measured and simulated data. Plots are automatically updated each time a measurement or simulation is performed.

Optimizing Model Parameters

Optimization of model parameters improves the agreement between measured and simulated data. An optimize transform whose *Extract Flag* is set to *Yes* is automatically called after any extraction that precedes it in the transform list.

Extracting Parameters

This section describes the general procedure for extracting model parameter data from the UCB MOSFET transistor. The general procedure applies to all types of parameters; differences between extracting one type and another are primarily in the types of instruments, setups, and transforms used. Also included in this section is information specific to DC and capacitance measurements and extractions.

Parameters are extracted from measured data taken directly from instruments connected to the inputs and outputs of the DUT. Using the extracted parameters simulated data can be generated by the simulator. Once measured and simulated data have been obtained, each data set can be plotted and the resulting Plots visually compared in the *Plot* window.

IC-CAP also extracts model parameters from simulated data. This capability is useful for creating a set of model parameters from the parameters of another model (parameter conversion) or for testing the accuracy of the extraction.

The general extraction procedure is summarized next, starting with the measurement process.

1 Install the device to test in a test fixture and connect the test instruments.

2 Ensure the test fixture, signal source and measuring instruments, and workstation are physically and logically configured for the IC-CAP system.

3 Load the model.

4 Select the DUT. In the DUT Parameters folder, enter the *W* and *L* device parameters for the selected DUT.

5 In the Macros folder, select the appropriate macro to enter the process parameters.

6 Select the setup.

7 Issue the *Measure* command.

8 Issue the *Extract* command.

9 Issue the *Simulate* command.

10 Display the results.

11 Fine tune the extracted parameters if needed by optimizing.

DC Measurement and Extraction

In DC parameter extraction, the extracted parameters are directly related to the geometries of the devices being tested. For a DUT to accurately extract DC model parameters, it must have the correct L (drawn or mask channel length) and W (drawn or mask channel width) device parameters. Before executing an extraction or simulation, edit each DUT to ensure the L and W parameters are correct.

Before starting the extraction, enter several process parameters. The most important of these is TOX. Determine TOX by reading the process information for the device, or by measuring the oxide capacitance; TOX is measured in meters. Enter its value directly in Model Parameters, or run the *init_parameters* macro. Also use the *init_parameters* macro to enter initial values for XJ, LD, and RS. These initial values can contribute to the accuracy of the extracted parameters. They are overwritten by new values when the XJ, LD, and RS are extracted during the extraction process.

Accurate results depend on the sequence of the extraction. Follow this DC extraction sequence.

- Extract the classical parameters from the large device. Because length and width effects are not critical for the device used in this step, the classical parameters can be extracted very accurately. These parameters are used for the remainder of the extractions.
- Extract parameters from a narrow device, in which length effects are not important but the width effect and width parameters are.
- Extract length parameters using a short channel device and the classical parameter data acquired in the first extraction. RS and RD parameters, which predominate in this device, are also extracted in this step.
- All of the parameters extracted are used to calculate the saturation parameters for the short channel device. The short channel device is used for this procedure because of the predominance of the saturation parameters.

Do all of the measurements, followed by all of the extractions, and finally, the simulations. Extraction usually provides a reasonable fit to the measured data, but you can optimize data to attain an increased level of accuracy. Execute the optimization after extracting the DC parameters for each setup.

To perform DC parameter measurements:

1 Choose **File > Open > Examples**. Select <*filename*>**.mdl** and choose **OK**. Open the model window.

When the model window appears you are ready to begin measurement and extraction operations.

NOTE P-channel and N-channel MOS extractions are handled the same. *pmos2.mdl* or *pmos3.mdl* files are used for P-channel extractions; *nmos2.mdl* and *nmos3.mdl* files are used for N-channel extractions.

2 Select the DUT **large**. Enter the values for L and W. To include the effect of WD, enter the following expression for W:

$$< value > - 2 \cdot \mathrm{WD}$$

where *value* is the drawn width and WD is defined as a model variable.

3 In Macros, select **init_parameters**. Enter the values for TOX, XJ, LD, and RS. Default values can be used by simply choosing **OK** in each dialog box.

4 Select the **idvg** setup and issue the **Measure** command.

5 Repeat these steps for narrow/idvg, short/idvg, and short/idvd.

To perform DC parameter extractions:

1 Select **large / idvg**. Select the transform **extract** and execute the selection to extract the LEVEL 2 classical parameters.

2 Repeat this procedure for narrow/idvg, short/idvg, and short/idvd.

All DC model parameters have now been extracted and their values are listed in Model Parameters.

Notes on DC Parameter Extraction

These procedures assume that the large device is large enough to make small geometry effects irrelevant. This condition exists when the device geometries are much larger than LD and WD. For a typical process, 50*50 microns should be sufficient. To improve accuracy, enter the approximate values of LD and WD in Model Parameters so they can be taken into consideration in the first extraction step. A more accurate value for each is produced by the second and third extractions.

When a very large device is not available and you cannot enter LD and WD, try the following:

1 Use the largest available device for the **large** setup and execute all four steps of the DC extraction.

2 Repeat the extraction sequence starting with the first step (you do not need to re-enter the parameters). The previously extracted parameters (particularly LD and WD) are used as the initial values.

To extract DC parameters when only one size of device is available, extract model parameters using the following sequence. This sequence does not extract geometry-dependent parameters but does extract a subset of parameters to fully model that size device.

1 Perform the **large/idvg** extraction to obtain the classical parameters.

2 Perform the **short/idvd** extraction to obtain the saturation parameters.

Enter the same L and W device parameters for both DUTs. The model can be reconfigured so that it has only one DUT with two setups, one similar to idvg and one similar to idvd. Copy the setup from large/idvg and the setup from short/idvd (copy complete setups so the appropriate extraction and optimization functions are included).

If you cannot determine the L and W for a single geometry device (as might be the case with a packaged transistor), set estimated values. The actual values are less important than the ratio between them. An incorrect ratio of *W/L* results in extraction of an unreasonable value for UO. In general, the mobility parameter UO should be set between 200 and 800. Start the extraction after setting the ratio of L and W to 1, then change the ratio of L to W to scale back the extracted value of UO.

Capacitance Measurement and Extraction

Capacitance parameters can be extracted before or after the DC parameters. The extraction requires that two different DUTs be measured; model parameters are extracted from the second DUT.

The extraction in the cbd1/cjdarea setup requires a single geometry to be measured and produces the parameters CJ, MJ, and PB. The extraction uses a transform set_CJ to find the initial zero bias value of CJ then uses optimization to obtain all three parameter values.

The extraction in the cbd2/cjdperimeter setup requires two geometries to be measured (one in the cbd1/cjdarea setup and one in the cbd2/cjdperimeter setup) that produces the parameters CJ, MJ, PB, CJSW, and MJSW—and therefore a more complete capacitance model.

The extract transform uses the MOSCV_total_cap function to simultaneously solve for the bottom area and sidewall capacitance parameters. To extract the capacitance contributions from the bottom area and the sidewall periphery the geometries must have different area-to-perimeter ratios. The device measured with the cbd1/cjdarea setup should have a high bottom area to perimeter area ratio and the device measured with the cbd2/cjdperimeter should have a low bottom area to perimeter area ratio.

Place the device to be measured into the test fixture. Ensure that the CMs (Capacitance Meters Units) connected to the device correspond to the same CMs in Table 7 for each of the next two measurements. Calibrate the capacitance meter before taking each measurement.

The extractions of the sidewall capacitance parameter sets use the measured data from both setups—measure both setups before performing the extraction.

1 In Macros, select **init_cap_parameters** and **Execute**.

2 Enter the drain area and perimeter information.

3 Connect the low terminal of the capacitance meter (CM low) to drain and connect the high terminal (CM high) to bulk.

4 Select **cbd1/cjdarea** and **Measure** to measure the first drain-bulk junction capacitance.

5 Repeat steps 3 and 4 for cbd2/cjdperimeter if both geometry sizes are being measured.

Perform the model parameter extractions.

1 If a single geometry was measured, select **cbd1/cjdarea**. If two different geometries were measured, select **cbd2/cjdperimeter**.

2 Choose **Extract**.

Optimization is usually not required for capacitance data.

Notes on Capacitance Parameter Extraction

The drain-to-substrate and source-to-substrate junction capacitances are modeled as a combination of the sidewall and bottom (area) capacitances. To extract the parameters for these capacitances, first measure capacitance against voltage on two different size capacitors. Then execute the extraction command using two setups: cjdarea and cjdperimeter. Execute cjdarea on a square-shaped capacitance with a small sidewall to bottom ratio, and cjdperimeter on a long, narrow junction with a large sidewall to bottom ratio.

Each p-n junction should be reverse-biased when measured. Extraction is performed by the MOSCV_total_cap function. The parameters CJ, MJ, CJSW, MJSW, and PB are calculated from a combination of the two measurements.

Before running the extraction, specify the area and perimeter of the capacitance. Enter these numbers by executing the init_cap_parameters macro. This sets the variables defined at the model level for the area and perimeter of the two DUTs. The parameters AD or AS (area) and PD or PS (perimeter) in the cbd1 and cbd2 DUTs are set by these variables.

Simulating

To simulate any individual setup choose Simulate with an active setup. Simulations can be performed in any order once all of the model parameters have been extracted.

IC-CAP provides a special function, MOSCVmodCBD, to speed up capacitance simulation in the cbd1 and cbd2 DUTs. This function models the simple pn junction capacitance and provides a fast simulation of the CBD capacitance. Use this function to execute a simulation by specifying the transform calc_mos_cbd_model in the setups for the two DUTs and Execute the Transform rather than issuing the Simulate command. For more information, refer to Chapter 9, "Using Transforms and Functions," in the *IC-CAP User's Guide*.

For more information on simulation, refer to Chapter 6, "Simulating," in the *IC-CAP User's Guide*.

Displaying Plots

Plots can be displayed from the Plots folder for the setup. To display plots issue the Plot Display command from a DUT to display the plots for all setups in that DUT. The plots use the most recent set of measured and simulated data. Viewing plots is an ideal way to compare measured and simulated data to determine if further optimization would be useful. For more information on plots, refer to Chapter 10, "Printing and Plotting," in the *IC-CAP User's Guide*.

Optimizing

The optimization operation uses a numerical approach to minimize errors between measured and simulated data. As with the other IC-CAP commands, optimization can be performed at either the DUT or setup level.

Optimization is typically interactive in nature, with the best results obtained when you specify the characteristics of the optimization function.

For more information, refer to Chapter 7, "Optimizing," in the *IC-CAP User's Guide*.

Extraction Algorithms

This section describes the extraction algorithms for the classical, narrow width, short channel, saturation region, and sidewall capacitance extractions.

Classical Parameter Extractions

This extraction calculates the classical model parameters UO, VTO, NSUB, and UEXP from the ID versus Vg measurement at varying bulk voltages on a large device. Select the gate voltage range to cover the cutoff as well as the linear region, including the mobility reduction range. The bulk should be biased at 0V as well as at values that cover the normal operating range of the device.

Parameters UO and VTO are first extracted from the Vb = 0 curve. To calculate these parameters, a least-squares fit is carried out to the maximum slope of the curve in the linear region. The parameter UEXP is calculated to fit the reduction in the slope of the same curve when higher gate voltages are applied. The parameter is calculated based on the specified value of UCRIT.

The combination of UO, UEXP and UCRIT has a redundant parameter. IC-CAP keeps the UCRIT fixed at its specified value and extracts UO and UEXP. An unreasonable value for UCRIT might result in an unexpected value for the mobility UO.

The same curve fitting is carried out on the curve with the largest absolute value of bulk voltage. The threshold voltage at this bias is then calculated from the intersection of this line. The parameter NSUB is calculated from the difference in the two threshold voltages.

Narrow-Width Parameter Extractions

This extraction calculates the narrow device parameters WD and DELTA from the Id versus Vg measurement. The setup and extraction are similar to the classical extraction. The threshold voltage (VTH) and Beta (effective mobility) are calculated using

least- squares fitting. The parameter WD is calculated from Beta and UO. The parameter DELTA is calculated from the shift in threshold voltage (the difference of VTH and VTO).

Short-Channel Parameter Extractions

This extraction calculates the short channel parameters LD and XJ from the Id versus Vg measurement. The setup and extraction are similar to the classical and narrow width. The effective Gamma (or effective NSUB) and Beta (effective mobility) are calculated using least- squares fitting. The parameter LD is calculated from Beta and UO. The parameter XJ is calculated from the change in Gamma (or NSUB).

The parameter XJ is the only parameter that controls the effect of channel length on the shift of threshold voltage due to bulk bias. This parameter is extracted by IC- CAP to fit the threshold shift and therefore its extracted value may not correspond to the metallurgical junction depth. In other words, XJ is an empirical (not a physical) parameter in this model.

Saturation Parameter Extractions

This extraction calculates the saturation parameters VMAX and NEFF from the Id versus Vd measurement. The measurement can be taken at a single gate voltage or at various gate voltages. Only the highest gate voltage curve is used in the extraction. Ensure the drain voltage sweep is sufficient to cover both the linear and saturation regions.

In this extraction, first the knee point or the saturation point is found from the shape of the curve for the maximum gate voltage. VMAX is calculated from the saturation point. NEFF is then calculated to fit the saturation portion of the curve.

Sidewall and Junction Capacitance Parameter Extractions

To accomplish total cv extraction (due to both bottom area and sidewall area), measure two DUTs using the same setup specifications. In these extractions, CJ and CJSW are calculated, then PB, MJ and MJSW are extracted.

CJ and CJSW Extractions

The values of CJ and CJSW are extracted from the measured capacitance data from the two different structures. The capacitors should have different ratios of their bottom area to sidewall area for best resolution of the equations.

The areas and perimeters used for the calculations are stored in the Model level variable table. The example MOS Model files provided with IC-CAP use variable names *AreaCap1*, *PerimCap1*, *AreaCap2*, and *PerimCap2*.

The capacitor in the *cjdarea* Setup has a capacitance dominated by the bottom area of the device. The capacitor in the cjdperimeter Setup has a capacitance whose perimeter area contribution is significant. The names in the variable table and in the DUT must match for the extraction to perform properly.

PB, MJ and MJSW Extractions

The parameter PB is extracted using the junction capacitance measurement not dominated by the sidewall effect. This is the DUT named cbd1 in the example MOS Model files.

The total capacitance is modeled by two equations that represent the bottom junction area and the sidewall junction area. The values of MJ and MJSW are obtained by simultaneously solving the two equations for total capacitance of each of the measured structures. An iterative method is used to obtain the built-in potential and grading factors.

HSPICE LEVEL 6 MOSFET Model

The general form of the Ids equation for the HSPICE LEVEL 6 MOSFET model is similar to the UCB MOS LEVEL 2 model. However, small geometry effects such as mobility reduction and channel length modulation are modeled differently. Also, the LEVEL 6 model can be used for modeling MOS transistors with ion-implanted channels due to its multi-level GAMMA capability.

The HSPICE MOS LEVEL 6 model is based on the ASPEC, MSINC, and ISPICE MOSFET model equations and has been enhanced by Meta-Software. Different versions of the model are invoked with the switch parameter UPDATE. There are more than 5 other switch parameters that are used for selecting different model equations. Refer to the *HSPICE User's Manual* [2] for more information on this model.

The IC-CAP LEVEL 6 model parameter extraction routines and configuration file are described in this section. Three extraction functions for this model are included in the IC-CAP function library. The configuration file, *hnmos6.mdl* supports a limited number and combination of parameters in the LEVEL 6 model. However, different parameter combinations can be supported by modifying the included optimization strategy. This configuration file can also be used for the HSPICE MOS LEVEL 7 model, provided that the PHI parameter is set to PHI/2 following the extraction.

NOTE Set the SIMULATOR variable to your version of HSPICE after loading the *hnmos6.mdl* configuration file into IC-CAP. Refer to Chapter 6, "Simulating," in the *IC-CAP User's Guide* for additional details on using HSPICE.

Model Parameters

The parameters used in the *hnmos6.mdl* example file are listed in Table 8. Six switch parameters are selected in the supplied configuration. The fixed parameter values are based on typical MOSFETs; they may need to be altered for certain devices.

An important feature of the HSPICE LEVEL 6 model is its multi-level Gamma capability. IC-CAP extraction routines support both single- and multi-level Gamma parameters extractions. If VBO is set to 0 before the Large IdVg extraction, only GAMMA is extracted. Otherwise, GAMMA, LGAMMA, and VBO are extracted. Optimization is necessary with the LEVEL 6 model for optimum agreement between measured and simulated data.

The IC-CAP Setup attributes for the LEVEL 6 model are listed in Table 9.

Table 8 HSPICE LEVEL 6 Parameters used in hnmos6.mdl

Switch Parameters	Fixed Parameters	Extracted Parameters
UPDATE = 1	BULK = 99	KU
ACM = 0	FDS = 0.9	MAL
CAPOP = 4	LATD = 0.2	MBL
MOB = 1	ESAT = 86.0E3	PHI
CLM = 3	KL = 0.05	VT
WIC = 1	KA = 0.97	GAMMA
	VSH = 0.7	LGAMMA (Optional)
	KCL = 1.0	VBO (Optional)
	MCL = 1.0	F1, LAMBDA, UB
	TOX (Input Parameter)	F3
	L (Input Parameter)	NFS
	W (Input Parameter)	LD or LDEL, WD or WDEL, RD, RS, XJ, DELTA, NWM, SCM, CJ, MJ, PB, CJSW, MJSW

Table 9 HSPICE LEVEL 6 Model Setup Attributes

DUT / Setup	Inputs	Outputs	Tranform	Function	Extractions
large / idvg	vg, vb, vd, vs	id	extract	MOSDC_lev6_lin_large	PHI, VT, GAMMA, LGAMMA, VBO, LAMBDA, UB, NFS
			optimize	Optimize	PHI, VT, GAMMA, LGAMMA, VBO F1, F3
			opt_NFS	Optimize	NFS
narrow / idvg	/ /	/ /	extract	MOSDC_lev6_lin_narrow	NWM, WD(EL), DELTA
			optimize	Optimize	NWM, WDEL
short / idvg	/ /	/ /	extract	MOSDC_lev6_lin_short	SCM, XJ, LD(EL)
			optimize	Optimize	SCM, XJ, LDEL, RD, RS
short / idvd	vd, vg, vb, vs	id	optimize	Optimize	KU, MAL, LAMBDA, MBL
cbd1 / cjdarea	vb, vd	cbd	set_CJ	Program	initial zero bias CJ
			extract	Optimize	CJ, MJ, PB
cbd2 / cjdperimeter	vb, vd	cbd	extract	MOSCV_total_cap	CJ, MJ, CJSW, MJSW, PB

Measurement

The measurement setups are identical to the UCB MOS LEVEL 2 and LEVEL 3 model example files. However, to obtain accurate GAMMA and LGAMMA parameters for ion-implanted devices, the measured data must clearly express the body effects. Therefore, the bulk voltage should be set broadly on the Large IdVg measurement. The following sequence for DC measurements is recommended:

1. Large IdVg
2. Narrow IdVg
3. Short IdVg
4. Short IdVd

Extraction and Optimization

All DC parameters are extracted and optimized with the DCExtraction macro. Alternately, extractions and optimizations can be performed interactively as described for the LEVEL 2 and LEVEL 3 MOSFET models. There is no extraction routine in the short IdVd setup for saturation region parameters. Instead, the parameters KU, MAL, MBL, and LAMBDA must be optimized. For certain devices it may be necessary to alter the optimization setup and default parameter values for accurate results.

References

1 A. Vladmirescu and Sally Liu. *Simulation of MOS Integrated Circuits Using SPICE2,* UCB/ERL M80/7, University of California at Berkeley.

2 *HSPICE User's Manual*, H92 Release, Meta-Software, Inc., 1992.

3 Using the BSIM Modeling Packages

Key Features of the BSIM3 and BSIM4 Modeling Packages 70
Data Structure inside the BSIM3 and BSIM4 Modeling Packages 72
DC and CV Measurement of MOSFET's for the BSIM3 and BSIM4 Models 77
RF Measurement 123
Extraction of DC/CV Parameters 137
Extraction of Parameters for the RF Models 176

This chapter discusses the measurement and extraction of parameters using the BSIM3 and BSIM4 Modeling packages developed by AdMOS. Both Modeling Packages use the same Graphic User Interface (GUI). Handling of the measurement and extraction tasks using one of the Modeling Packages is identical and therefore only needs to be described once.

There are a few exceptions: BSIM3 and BSIM4 use different model flags and parameter sets. Therefore, the initialization procedure is somewhat different. This is the reason that you will find — on places where necessary — screen shots from both modeling packages.

Model specific parts are located in Chapter 4, "BSIM3v3 Characterization" and Chapter 5, "BSIM4 Characterization." Inside those chapters you will find some theoretical aspects for each of the models. Links are provided to bring you from this chapter to the respective chapters and vice versa.

Key Features of the BSIM3 and BSIM4 Modeling Packages

- The new graphical user interface in Agilent's IC-CAP enables the quick setup of tests and measurements followed by automatic parameter extraction routines.
- A new data management concept allows powerful and flexible handling of measurement data using an open and easy data base concept.
- The powerful extraction procedures can be easily adopted to different CMOS processes. They support all possible configurations of the BSIM3 and BSIM4 models.
- Quality assurance procedures are checking every step in the modeling flow from measurements to the final export of the SPICE model parameter set.
- The fully automatic generation of HTML reports is included to enable web publishing of a modeling project.
- The modeling package supports SPICE3e2 and major commercial simulator formats such as HSPICE, Spectre, and Agilent's ADS.

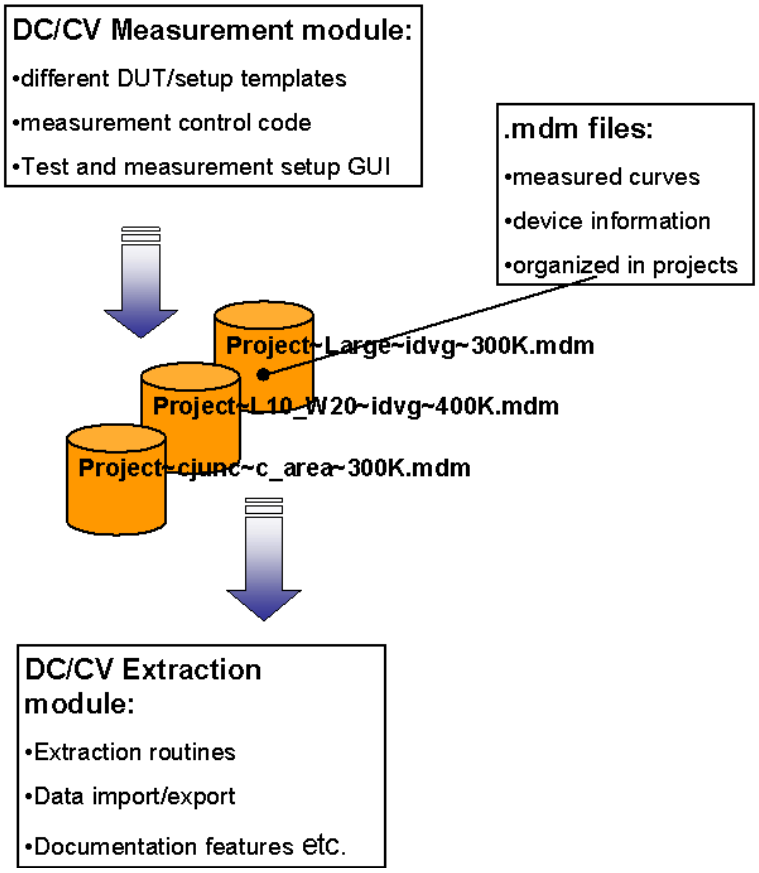
The Modeling Packages Support Measurements on

- Single-finger (normal) transistors
- Parasitic diodes
- Capacitances:
  - Oxide
  - Overlap
  - Bulk-drain, source-drain junction
  - Intrinsic
- RF multifinger transistors

The Modeling Package Supports Extractions for

- Basic transistor behavior
- Parasitic diodes
- Capacitances
- RF behavior (S-parameters)

Data Structure inside the BSIM3 and BSIM4 Modeling Packages



The Modeling Packages are using a totally different, more advanced data storing concept compared to former modeling products inside IC-CAP. The drawback of recent modeling products was that measured data was always stored in model files together with transforms, macros, plot definitions, and so on.

This method has two major disadvantages:

- The additional information is stored *n* times and is therefore highly redundant.

- The combination of data and code makes it very difficult to introduce updates to the code.

Now, the new architecture of the BSIM3 and BSIM4 Modeling Packages overcome these disadvantages.

The measurement module contains all measurement related items like DUTs/Setups to perform measurements and setup of test and measurement conditions. The measured data is stored together with device information like gate length, pin numbers of a switch matrix used, and so forth in IC-CAP *.mdm* data base format. These *.mdm* files are organized as projects which can be identified by project name.

Now, the extraction module extracts the necessary data from stored *.mdm* files to perform model parameter extraction and visualization of measured and simulated results. In addition, this method allows the generation of new data representations where the scalability of a model can be easily verified.

Files resulting from Measurement and Extraction using the Modeling Packages

This section describes the files resulting from measurement and extraction of MOS devices using the BSIM3 and BSIM4 Modeling Packages. The following table shows how BSIM3/BSIM4 files names are being used. The bold printed words inside the left most column names the task the files are being used for and the normal printed names are the appropriate file names used to store the files for a specific project. The columns marked DC and RF list the task performed by each file. For example, measure+extract in the column DC means that this file is used in measurement and extraction of DC parameters.

NOTE The following characters are excluded from use in file or project names: “ / \ , . ; : * ? ~ % $ ’ ä Ä ö Ö ü Ü as well as “empty space”.

Table 10 Data Structure of BSIM3 and BSIM4 Modeling Packages

File Usage	DC	RF	Comment
Project search			
*~dc_idvg~*K.mdm	measure + extract		
*~rf_s_dut~*K.mdm		measure + extract	
Settings			
project_name~dc_meas~settings.set	measure + extract		
project_name~rf_meas~settings.set		measure + extract	
project_name~lwc(model name)~settings.set	extract	extract	
Boundaries			
*~lwc(model name)~boundaries.set	extract	extract	default for export, * = *project_name*
MPS			
project_name~lwc(model name).mps	measure + extract	measure + extract	for scaled model
project_name~DUT_name~lwc(model name).mps		extract	for single model
LIB			
project_name~DUT_name~Simulator~lwc(model name).lib		extract	for single model
project_name~Simulator~lwc(model name).lib	extract	extract	for scaled model

Table 10 Data Structure (continued)of BSIM3 and BSIM4 Modeling Packages

File Usage	DC	RF	Comment
Logfile			
project_name~lwc(model name)~log_fail.txt	extract	extract	
MDMs			
project_name~DUT_name~dc_idvg~TempK.mdm	measure + extract		
project_name~DUT_name~dc_idvd~TempK.mdm	measure + extract		
project_name~DUT_name~c_bd_area~TempK.mdm	measure + extract		
project_name~DUT_name~c_bd_perim~TempK.mdm	measure + extract		
project_name~DUT_name~c_bd_perim_gate~TempK.mdm	measure + extract		
project_name~DUT_name~c_bs_area~TempK.mdm	measure + extract		
project_name~DUT_name~c_bs_perim~TempK.mdm	measure + extract		
project_name~DUT_name~c_bs_perim_gate~TempK.mdm	measure + extract		
project_name~DUT_name~ c_d_g~TempK.mdm	measure + extract		
project_name~DUT_name~ c_g_ds~TempK.mdm	measure + extract		
project_name~DUT_name~ c_g_dsb~TempK.mdm	measure + extract		
project_name~DUT_name~di_bd_area~TempK.mdm	measure + extract		
project_name~DUT_name~di_bd_perim~TempK.mdm	measure + extract		

Table 10 Data Structure (continued)of BSIM3 and BSIM4 Modeling Packages

File Usage	DC	RF	Comment
project_name~DUT_name~di_bd_perim_gate~TempK.mdm	measure + extract		
project_name~DUT_name~di_bs_area~TempK.mdm	measure + extract		
project_name~DUT_name~di_bs_perim~TempK.mdm	measure + extract		
project_name~DUT_name~di_bs_perim_gate~TempK.mdm	measure + extract		
project_name~DUT_name~rf_s_dut~TempK.mdm		measure + extract	
project_name~DUT_name~rf_id_bias_points~TempK.mdm		measure	
project_name~DUT_name~rf_idvd~TempK.mdm		measure + extract	
project_name~DUT_name~rf_s_open~TempK.mdm		measure	
project_name~DUT_name~rf_s_short~TempK.mdm		measure	
project_name~DUT_name~rf_s_through~TempK.mdm		measure	
HTML			
PathHTML/index.htm	extract	extract	Start file
PathHTML/*.htm	extract	extract	
PathHTML/menu.js	extract	extract	File structure
PathHTML/imgmenu/*	extract	extract	Pictures for file structure
PathHTML/setups/*.htm	extract	extract	Measurement Setups
PathHTML/results/*.htm	extract	extract	Pages with results
PathHTML/results/*.txt	extract	extract	Parameter set for displaying HTML
PathHTML/results/images/*.gif	extract	extract	Images
PathHTML/results/imgzoom/*.gif	extract	extract	Zoomed Images

DC and CV Measurement of MOSFET's for the BSIM3 and BSIM4 Models

This section provides information to make the necessary measurements of your devices. It will provide information on features of the BSIM3 and BSIM4 Modeling Packages and how to use the graphic user interface (GUI). For hints on how to measure and what to measure using the right devices, see Chapter 4, "BSIM3v3 Characterization" and Chapter 5, "BSIM4 Characterization."

The GUI is opened by double clicking the BSIM3 or BSIM4 Icon which appears in the IC-CAP/Main window after you open one of the example files. To open an example file, click *File* > *Examples* > *model_files* > *mosfet* > *bsim3* (or *bsim4*) then select a Measure or Extract model file. Figure 1 shows all four files in one IC-CAP/Main window, using BSIM3 as an example.

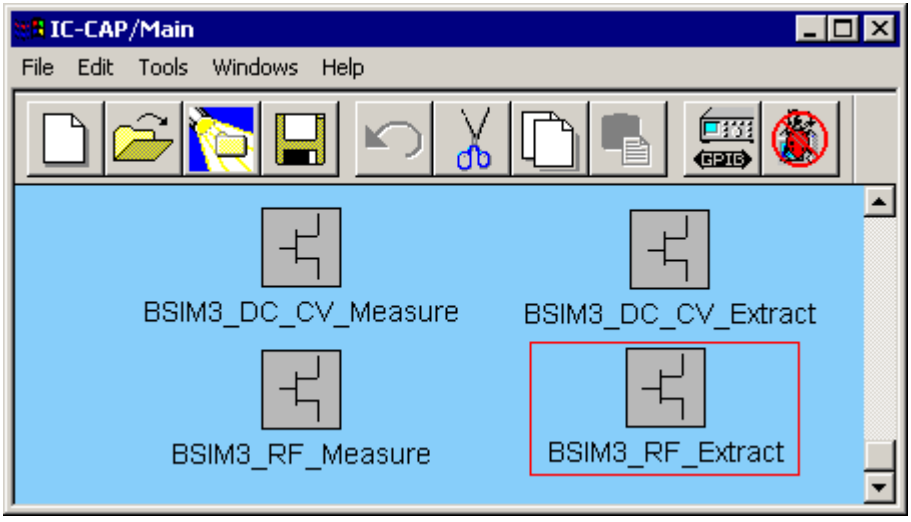


Figure 1 Starting the BSIM3 GUI from IC-CAP/Main window

After you have double clicked the icon, the GUI window of the BSIM3 (or BSIM4) Modeling Package (Figure 2) appears on your screen.

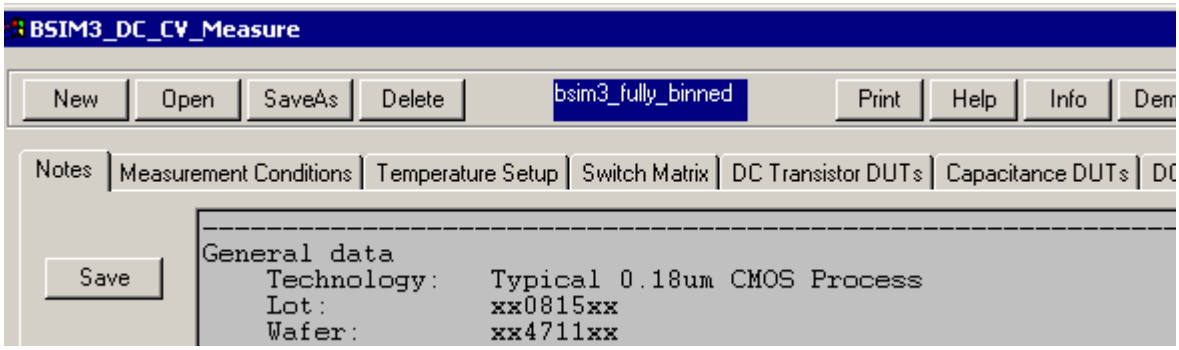


Figure 2 Part of the Graphic User Interface for the BSIM3 and BSIM4 Modeling Packages

The top row of the GUI shows a group of buttons on the left side that enable you to create a *New* project, to *Open* an existing project, to *SaveAs* or to *Delete* projects. You will be prompted before the selected action takes place.

The project name appears in the middle of the top row. In Figure 2, the project name is “bsim3_fully_binned” and is shown on a blue background.

The right side of the GUI’s top row shows a *Print* button, which opens a dialog box. In this dialog box, enter the command line for your specific printing device and choose *OK*. The folder will be printed.

NOTE On Windows operating system, the command line is print /d:<printer name>. For example, if the printer is connected to a server named MYFS1 and the printer is named MY0017, type:

print /d:'\\MYFS1\MY0017'

NOTE If you don’t enter a printer command, the output will be redirected to the IC-CAP/Status window.

The next button in the top row - *Help* - opens up the online manual - the file you are viewing right now.

In contrast to the function of this button, you'll find a *Help* button on each folder's lower left corner, which also opens the manual at specific locations describing the task to be performed using that folder.

There are in depth hints for the task, for example, which device geometries to use or how to connect the instrument to the device under test to get the best extraction results from your measurements. You will find links that bring you to Chapter 4, "BSIM3v3 Characterization" and Chapter 5, "BSIM4 Characterization." Use your Browsers *Back* button to return to the location you were at before following the link.

The *Info* button, which is located to the right of the help button, gives you some information about the creators of the BSIM3 and BSIM4 Modeling Packages.

The fourth button in this group is the *Demo* button. Use this button to explore the BSIM Modeling Packages features without starting actual measurements. This means, all measurement device drivers are disabled. Therefore, no measurement is possible in demo mode! This is also a convenient way to create a project without needing a measurement license!

There is a button far to the right of the top row to *Exit* the BSIM Modeling Packages.

Below the top row of buttons you will find a row of eight folders. Basically, each folder is assigned to a specific task in the measurement process. They are intended to be parsed from left to right, but you are not bound to that order. Some entries into one or the other folder will change settings on another folder. For the new user, we recommend to process the folders in the order from left to right.

Each of the following sections describe one folder of the GUI. BSIM3 and BSIM4 folders are usually equal to each other. In places where they differ - *Initialize*, for example - they will be described one after the other.

Project Notes

The notes folder is provided to store notes you take on a specific project, Figure 3 shows the notes folder. You can enter general data like technology used to produce this wafer as well as lot, wafer, and chip number. There is a field to enter the operator's name and the date the measurement was taken. Space has been provided to enter notes on that project.

The notes you enter are saved under the project name in the middle of the top row using the *Save* button to the left of this folder. In our example, this project is called "bsim3_fully_binned".

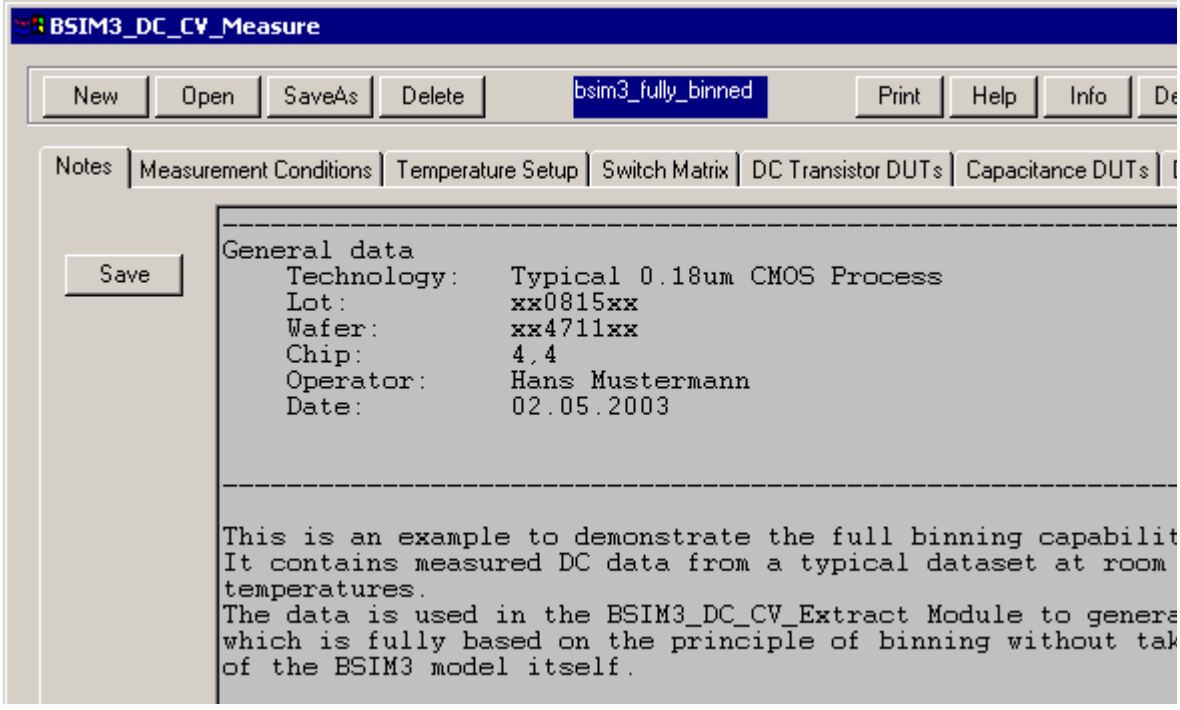


Figure 3 Notes folder of the BSIM3 and BSIM4 Modeling Package's GUI

DC Measurement Conditions

The next step in the modeling process is to set up measurement conditions for different measurement tasks like DC, CV, or diode measurements.

This folder is designed for easy setup of conditions for DC Transistor and Capacitance as well as DC Diode measurements. Figure 4 shows the folder used for setting up measurement conditions. On the left side of the folder, you will find a button to *Save* your setup under the name of the project in the middle of the top row. In this example, the name is “bsim4_for_experts”.

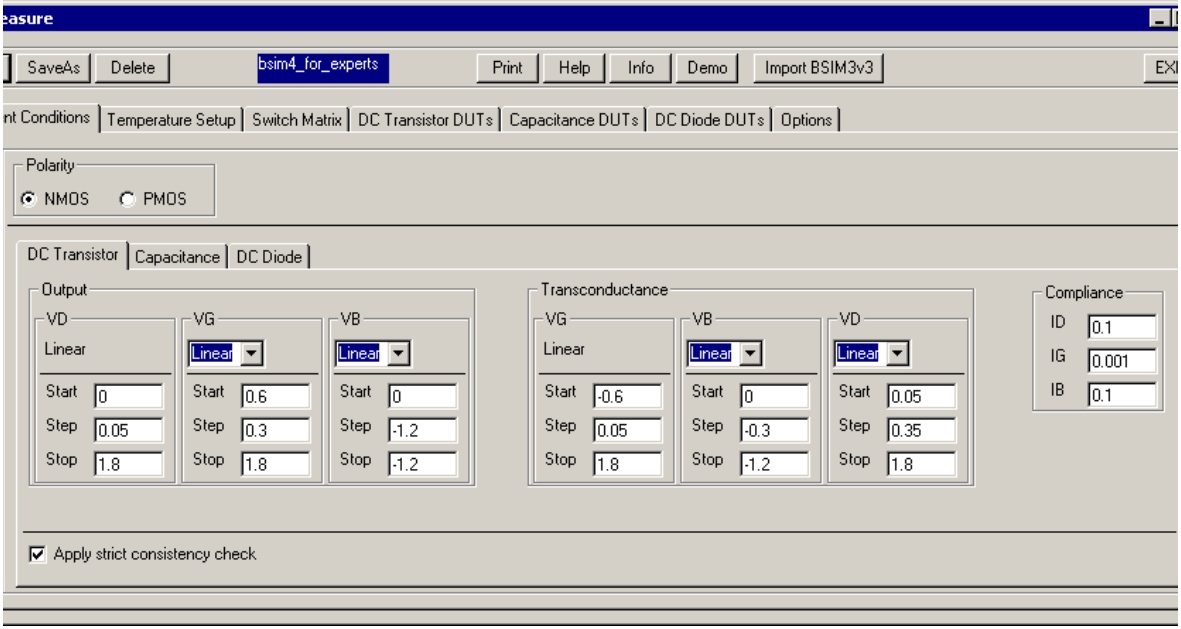


Figure 4 Measurement Conditions Folder

The measurement conditions folder is divided into sections to enter the polarity of the devices to be measured and to define the conditions for DC Transistor, Capacitance, and DC Diode measurements.

There are three subfolders, one for the DC Transistor measurements, one for Capacitance measurements, and one for Diode measurements. It is now possible to switch from a *Linear* voltage sweep with Start/Stop/Step values to be entered to a *List* of values, if you would like to measure discrete voltage values.

NOTE If you switch from *Linear* to *List* mode, at first only a number of points of this list is displayed. To get the list itself, click into the number field and press "Return". The list will appear!

Below you will find a check box to activate a consistency check. If checked, the measured data will be checked for errors, see "Consistency Check of DC measurement data for multiple measured devices" on page 234 for details.

On the *Capacitance* measurement subfolder inside the BSIM3/4 Modeling Packages, it is now possible to switch the "High" input of the CV meter between the nodes by activating the field in front of the Transistor terminal, inside the area *Connect "High" to*, see Figure 5.

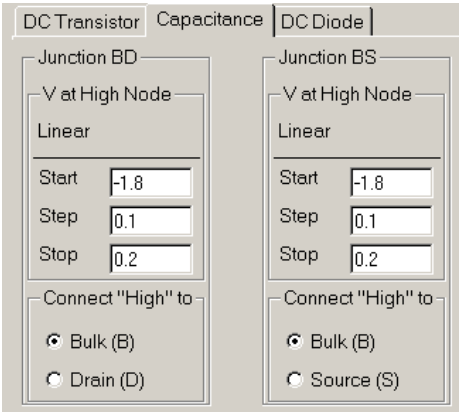


Figure 5 Part of the *Measurement Conditions* Folder, *Capacitance* measurement subfolder

See the sections on “DC Transistor DUTs” on page 92, “Capacitance DUTs” on page 108, and “DC Diode DUTs” on page 116, respectively, for background information on connecting the source-measurement units (SMU’s) as well as the CV instrument to the devices under test.

Polarity

There are polarity boxes to specify whether you are measuring NMOS or PMOS devices. Select NMOS or PMOS. Our example shows the measurement of NMOS devices.

DC Transistor

- Output (I_D=f(V_D))

Here you specify the stimulus voltages used for measuring the output characteristic of your devices. There is a choice between a *Linear* sweep and a *List* of discrete voltage values, where you can enter a number of points and their respective value. For *Linear* sweep mode, you define *Start*, *Step*, and *Stop* voltages for drain, gate, and bulk nodes, respectively. Figure 6 shows a typically measured output characteristic of a MOSFET.

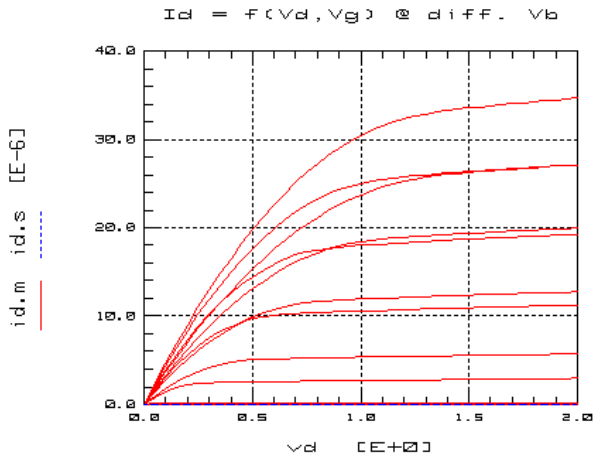


Figure 6 Output diagram of a MOSFET

- Transconductance (I_D=f(V_G))

This part of the measurement conditions folder is designed for transfer diagram measurements. Again, there is a choice between a *Linear* sweep and a *List* of discrete voltage values, where you can enter a number of points and their respective value. For *Linear* sweep mode, you specify *Start*, *Step*, and *Stop* voltages for gate, bulk, and drain nodes. Stop value of drain voltage is set to a fixed value in order to measure the relevant range of voltages for proper extraction of the parameters used to model this device behavior.

Figure 7 shows the typical form of a transconductance diagram.

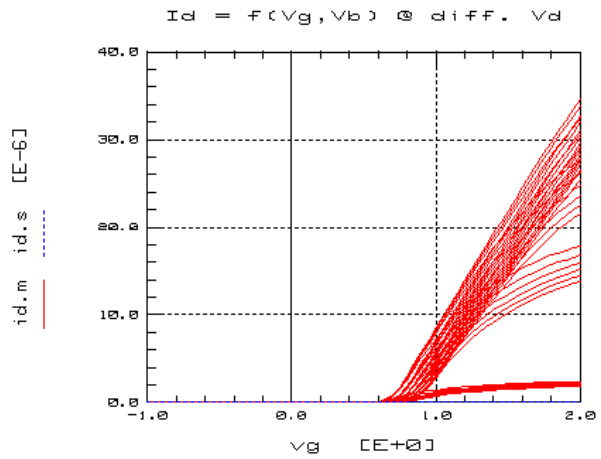


Figure 7 Transconductance diagram

If you change the settings of the diagram in the figure above, one of the effects appearing in submicron semiconductor devices becomes visible. The following figure shows a typical transconductance diagram using a logarithmic y-axis to show the influence of the GIDL (gate induced drain leakage) effect on transistor behavior.

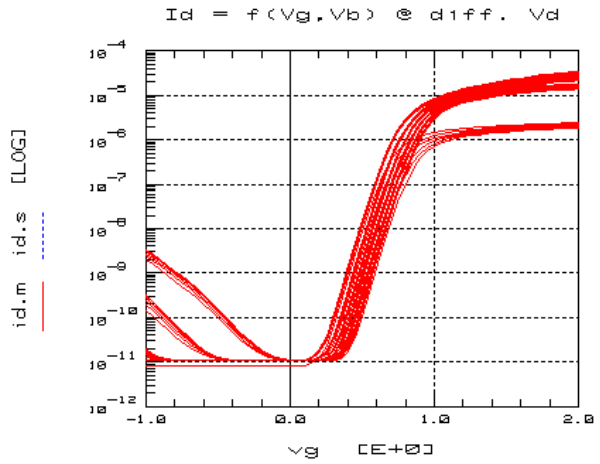


Figure 8 Transconductance diagram showing GIDL effect

Capacitance

This section is used to define capacitance measurement conditions for junction, oxide/overlap and intrinsic capacitances. See Figure 9 for a definition of capacitances on a MOSFET.

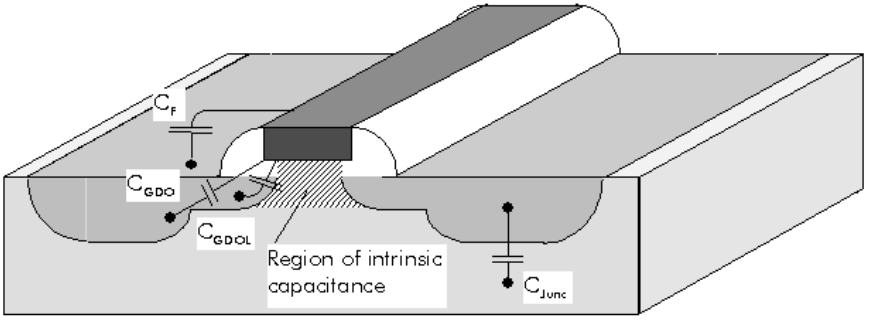


Figure 9 Definition of capacitances on a MOSFET

- Junction

  Enter the *Start*, *Step*, and *Stop* values for the bulk voltage used to measure junction capacitance of drain/bulk and source/bulk junctions. If you are using the BSIM4 model, you can measure the BS and BD junctions separately.

- Oxide/Overlap

  Provide values for *Step* and *Stop* voltage used in measuring gate/source, gate/drain, and gate/bulk overlap capacities as well as oxide fringing capacity.

- Intrinsic

  Here you can specify values for drain and gate voltages used to measure intrinsic capacitances.

DC Diode

This part of the measurement conditions folder is used to define DC measurements on source/drain-bulk diodes. To be defined is the *Start*, *Step*, and *Stop* voltage for the SMU connected to the drain node.

If you are using the BSIM4 model, you are able to measure BS and BD Diodes separately and you can specify if you would like to measure the current at the Bulk node or at the Drain or Source node.

Figure 10 shows a cross section of a MOSFET with the source-bulk diode shown.

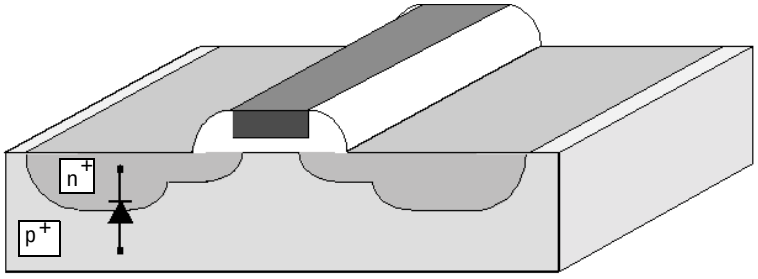


Figure 10 Measurement of source/drain-bulk diodes

NOTE Be sure to set the *Compliance* values for your measurements by entering the required values into the fields provided!

Temperature Setup

This folder is intended to define measurements at specified temperatures. Basically, the measurement of all DUT's is to be performed at SPICE default temperature TNOM, which is set to 300 Kelvin (27° Celsius). This temperature can not be deleted. Figure 11 shows the Temperature Setup folder.

NOTE Don't forget to enter the actual temperature during measurement of the devices.

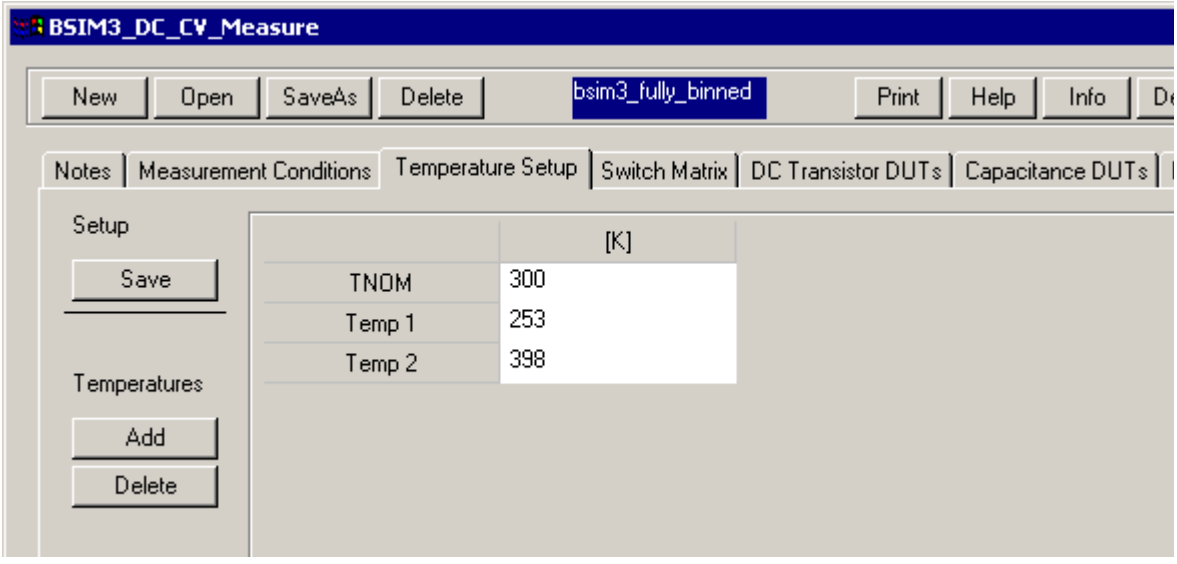


Figure 11 Temperature Setup folder

Using the buttons provided on the left side of the folder, you can *Add* new temperatures. Enter the desired temperature into the dialog box. Please be sure to enter the appropriate value in degrees Kelvin (K).

If you would like to *Delete* a measurement temperature, you will be prompted for the temperature to be deleted. If there is a file containing measured data for this temperature, the data file will be deleted if you choose *OK* on the prompt dialog (see the following figure).

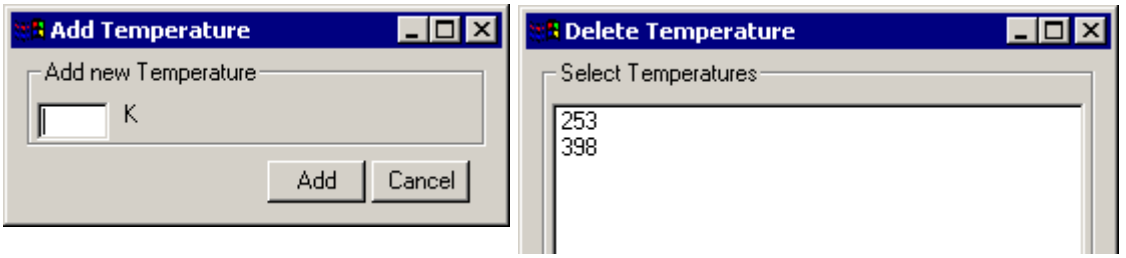


Figure 12 Dialog boxes to add and to delete temperatures (and measured data files, where applicable!)

NOTE You cannot delete the nominal temperature TNOM!

Adding new measurement temperatures results in adding a new column for each of these temperatures in the three DUTs folders (DC Transistor, Capacitance, and DC Diode).

Any changes on the Temperature Setup folder must be saved prior to selecting another folder.

Switch Matrix / Wafer Prober

Within this folder, which is shown in Figure 13, you select the kind of measurement you are using a switch matrix for. There are three options: Use a switch matrix for *DC*

Transistor Measurements, for *Capacitance Measurements*, and for *Diode Measurements*. You can select any one or more than one by checking the appropriate fields(s).

NOTE

If you are not using a switch matrix, leave all three check boxes unchecked. In this case, you do not have terminal assignment columns on the DC Transistor DUTs folder. Instead, you determine the connections by wiring the appropriate SMU to the desired transistor terminal.

NOTE

Assignments have to use SMU1....SM4. This assignment is done inside the hardware setup of IC-CAP. Usually, the default of the appropriate DC-CV-Analyzer is SMU1...4. In rare cases, such as the Agilent 4142 for example, the default SMU number corresponds to the number of the slot, a module is inserted into the instrument. If your 4142 uses 4 SMU's at slot No. 1, 3, 5, 6, the default names of the SMU's are SMU1, SMU3, SMU5 and SMU6. You must change this default names to reflect SMU1, SMU2, SMU3 and SMU4 to properly communicate with the BSIM3/4 modules.

NOTE

To change or enter the names of the Source-Measurement-Units (SMU's), go to the folder *DUTs/Setup*, sub folder *Measure/Simulate* in IC-CAP/Main window. The different inputs/outputs are to be configured there.

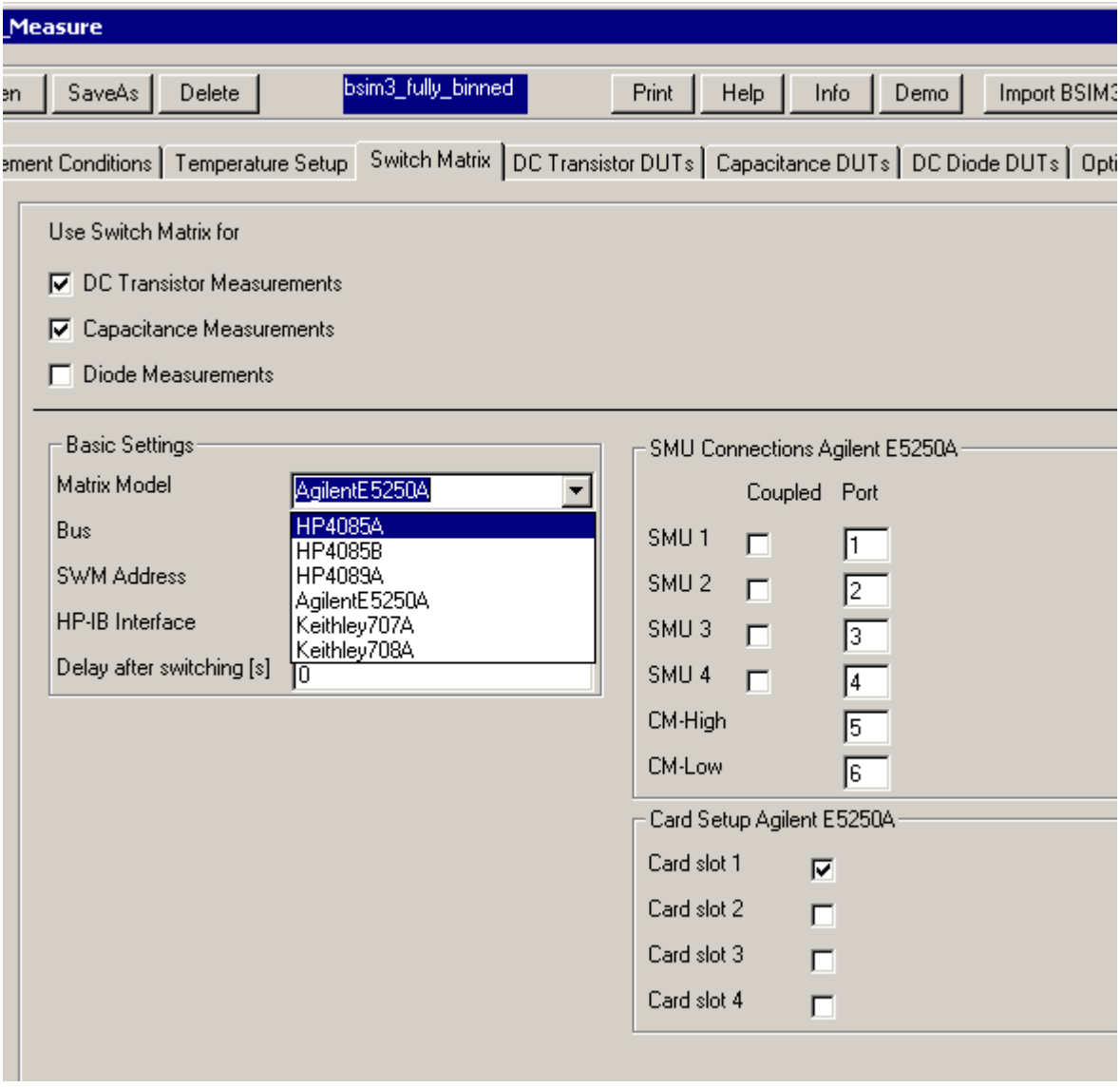


Figure 13 Defining the use of a switch matrix for measurements

The *Basic Settings* provide choice of several different Matrix Models, which are supported by IC-CAP. Type the appropriate Bus and GPIB address of the Switch Matrix (SWM Address; 22 in our example) as well as the GPIB-Interface name. See the IC-CAP *Reference* manual for a complete description of the GPIB settings for the switch matrix to be used. Our example shows the use of an Agilent E5250A matrix model. For this type of instrument, you have to define which port is connected to what SMU or C meter input pin and which slot is equipped with a card.

Again, you have to save your changes prior to leaving this folder.

The actual pin connections are entered into the folder selected to use a switch matrix (one or more of the *DC Transistor DUTs*, *Capacitance DUTs, Diode DUTs* folder, or all of them). For example, if you've selected *DC Transistor measurements* to use with a switch matrix, you must enter the switch matrix pin numbers in the fields below the node names of the transistors to be measured on the *DC Transistor DUTs* folder. This might be especially useful if you would like to make series measurements on wafers using a probe card (e.g., for quality control).

NOTE

For automatic measurements, macros are available. These macros allow you to make automatic series measurements of complete dies or arrays. They are created for automatic measurements with or without heated chucks.

For example, open the IC-CAP model for BSIM3_DC_CV_Measure (Right hand mouse button: Edit, with the pointer set over the DC_CV_Measure model in the IC-CAP/Main window): Under the folder macros you will find one called *Example_Wafer_ Prober*.

You will find one inside the *$ICCAP_ROOT/examples/model_files/mosfet/BSIM3/examples/waferprober* directory as well as inside the *.../BSIM4/examples/waferprober* directory named

prober_control.mdl. Please use one of these macros or model files and tailor it to your needs. There are readme sections to explain the steps to be taken inside the macros.

The automatic measurement of model curves using the macro described works without involving the GUI.

DC Transistor DUTs

The DC Transistor DUTs folder is used to enter DUT names, geometries, and connections to the appropriate DUTs. Since there are differences between BSIM3 and BSIM4, screen shots of both are provided here. The BSIM4 model allows stress effect modeling, which is not possible in BSIM3. Therefore, all stress effect parameters are only used inside the BSIM4 Modeling Package and are present only there!

Figure 14 shows the DC Transistor DUTs folder used.

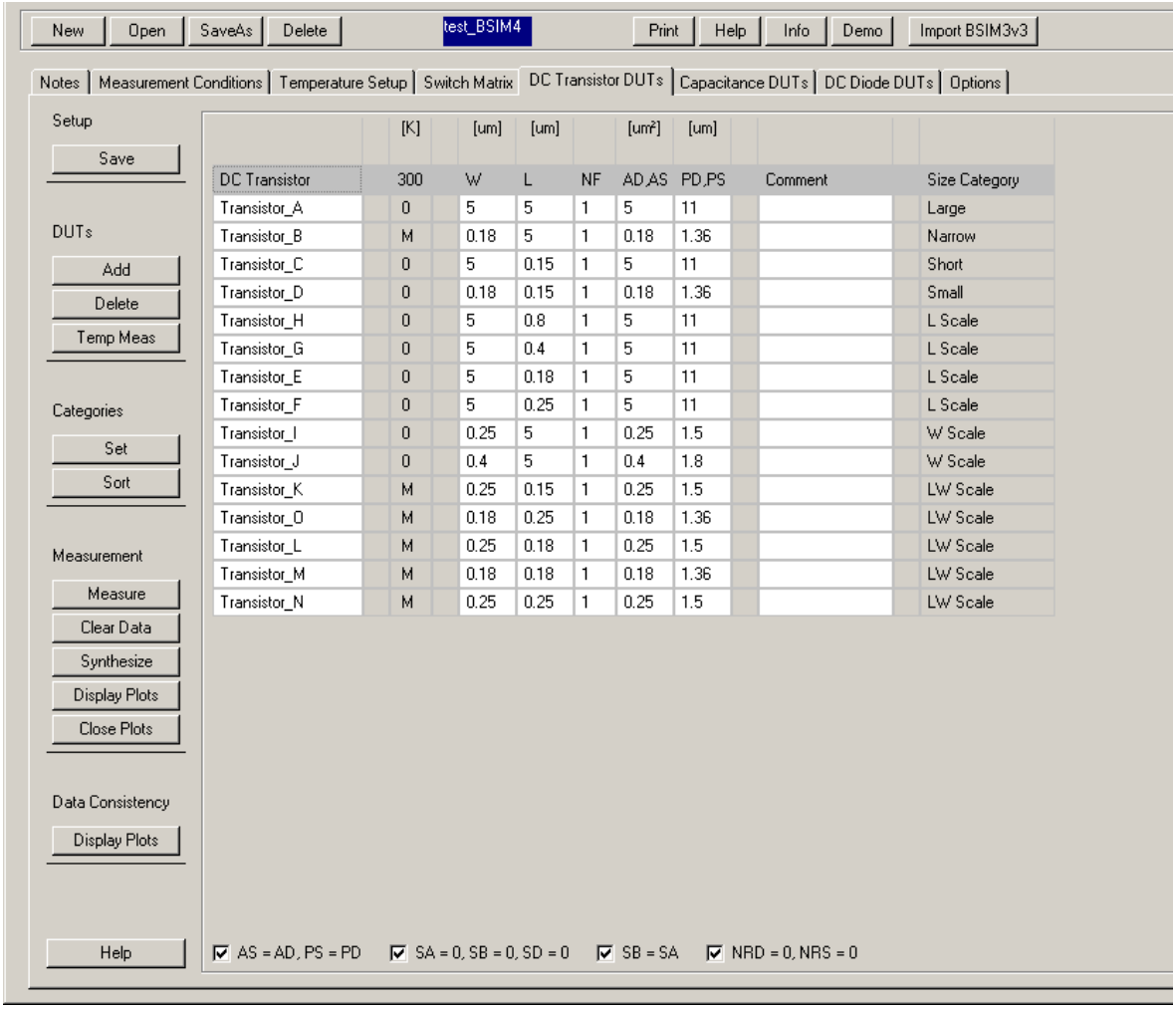
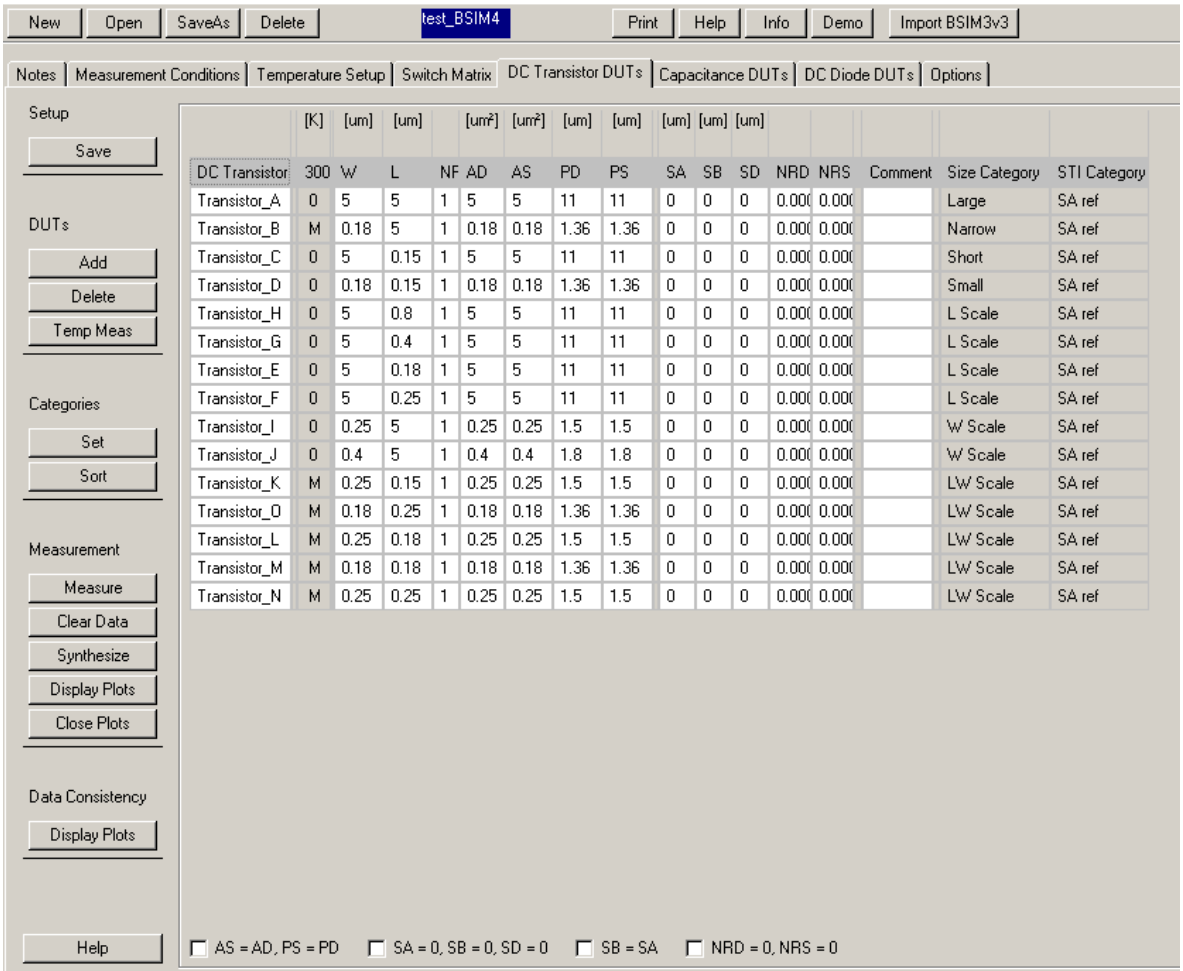


Figure 14 DC Transistor DUTs folder (default settings when starting a new project)

At the bottom of the BSIM_DC_CV_MEASURE module you will find fields to set BSIM4-specific values. You are able to use different area and perimeter values as well as Number of Squares for the Drain and Source regions of the transistors to be measured (AS, AD, PS, PD, NRD, NRS).

Further on it is possible to set stress effect parameters SA, SB, SD. See “Stress Effect Modeling” on page 372 inside the Chapter BSIM4 Characterization for details.

If you deselect one of the fields at the bottom of the DCTransistor DUTs folder, additional columns appear, as shown in Figure 15.



	[K]	[um]	[um]		[um²]	[um²]	[um]	[um]	[um]	[um]	[um]				
DC Transistor	300	W	L	NF	AD	AS	PD	PS	SA	SB	SD	NRD	NRS	Comment	Size Category	STI Category
Transistor_A	0	5	5	1	5	5	11	11	0	0	0	0.000	0.000		Large	SA ref
Transistor_B	M	0.18	5	1	0.18	0.18	1.36	1.36	0	0	0	0.000	0.000		Narrow	SA ref
Transistor_C	0	5	0.15	1	5	5	11	11	0	0	0	0.000	0.000		Short	SA ref
Transistor_D	0	0.18	0.15	1	0.18	0.18	1.36	1.36	0	0	0	0.000	0.000		Small	SA ref
Transistor_H	0	5	0.8	1	5	5	11	11	0	0	0	0.000	0.000		L Scale	SA ref
Transistor_G	0	5	0.4	1	5	5	11	11	0	0	0	0.000	0.000		L Scale	SA ref
Transistor_E	0	5	0.18	1	5	5	11	11	0	0	0	0.000	0.000		L Scale	SA ref
Transistor_F	0	5	0.25	1	5	5	11	11	0	0	0	0.000	0.000		L Scale	SA ref
Transistor_I	0	0.25	5	1	0.25	0.25	1.5	1.5	0	0	0	0.000	0.000		W Scale	SA ref
Transistor_J	0	0.4	5	1	0.4	0.4	1.8	1.8	0	0	0	0.000	0.000		W Scale	SA ref
Transistor_K	M	0.25	0.15	1	0.25	0.25	1.5	1.5	0	0	0	0.000	0.000		LW Scale	SA ref
Transistor_O	M	0.18	0.25	1	0.18	0.18	1.36	1.36	0	0	0	0.000	0.000		LW Scale	SA ref
Transistor_L	M	0.25	0.18	1	0.25	0.25	1.5	1.5	0	0	0	0.000	0.000		LW Scale	SA ref
Transistor_M	M	0.18	0.18	1	0.18	0.18	1.36	1.36	0	0	0	0.000	0.000		LW Scale	SA ref
Transistor_N	M	0.25	0.25	1	0.25	0.25	1.5	1.5	0	0	0	0.000	0.000		LW Scale	SA ref

Figure 15 Additional columns for stress effect modeling added in BSIM4

Now you are able to enter STI-related parameters, refer to Figure 18 for details.

For your convenience, there are predefined DUTs on this folder. You can either use those predefined DUTs, only adjusting names, device geometries, connections and so on, or you can delete existing DUTs and add your own.

- Choose *Add* on the left side of the folder. You will be prompted for the number of DUTs to add. Enter the desired number and choose *Add*. It is also possible to use the arrows to increase or to decrease the number of DUTs to add. New lines are added according to the number you've entered.

For each line, enter a name for the DUT, gate length and width (L, W), drain and source areas (AD, AS), perimeter length of drain and source (PD, PS), and the number of device fingers (NF) of the transistor to be measured. If modeling stress effects in BSIM4, enter SA, SB, and SD as well. See Figure 16 as well as Figure 17 for details on device geometry, respective Figure 18 for details on STI modeling parameters.

NOTE

Remember, all geometries are to be given in microns (µm).

Geometries

Shown in the following two figures are views of MOSFET's, where you can find the geometries required by the BSIM3 and BSIM4 modeling packages.

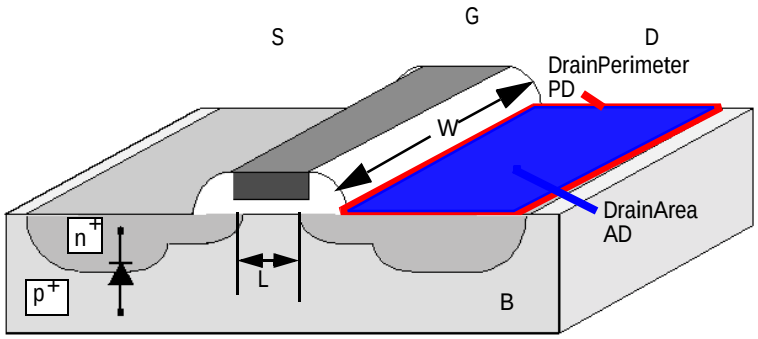


Figure 16 .Cross section of a MOSFET showing device geometries

You are not bound to an order of entry. This means, it is not required to begin with the large transistor, the short transistor or the narrow one. Just type in the geometries into each line as you like.

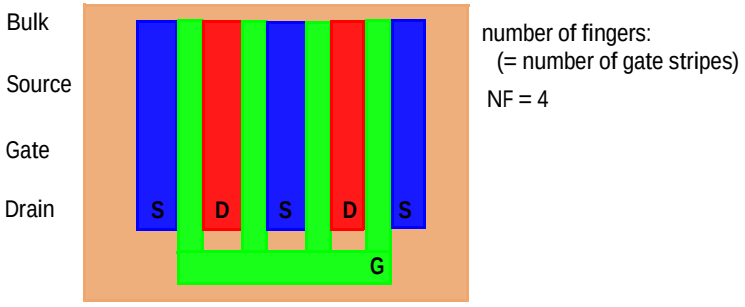


Figure 17 Top view of a multifinger MOSFET

In addition to the standard BSIM4 definitions, columns for SA, SB and SD are available, see Figure 18 for a definition of the parameters used in shallow trench isolation modeling.

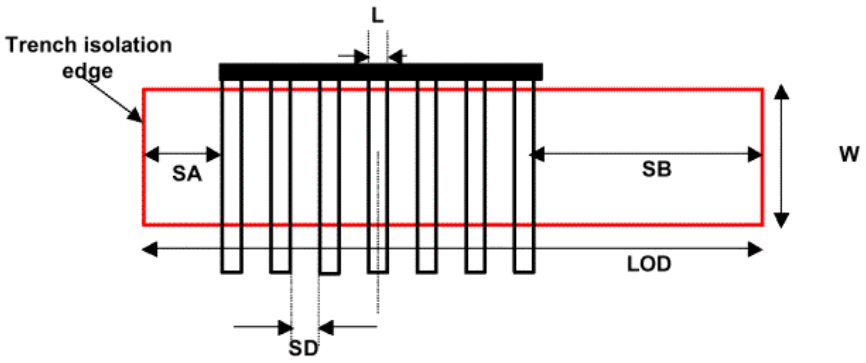


Figure 18 Shallow Trench Isolation related parameters

Moreover, each transistor is assigned to two different categories as is described in the following section.

Categories

The DC Transistor DUTs folder shows one or two category columns: The first one is the size category, the second one is the STI category.

- Size category (applicable for BSIM3 *and* BSIM4): determines the properties of a transistor regarding channel length (L) and width (W).
- STI category (applicable only for BSIM4): determines the properties of a transistor regarding the actual value of SA/SB. Mainly, this category defines, whether a device belongs to the reference values SAREF, SBREF or has SA, SB values, which are different from the reference values.

NOTE Devices at SA=SAREF and SB=SBREF are used to determine all other model parameters except the STI related parameters.

There are buttons to *Sort* the entries into an order or to *Set* the size category of your devices manually. See "Transistors for DC measurements" on page 293. You will find a graph of

recommended device geometries in the form of a diagram W over L of the transistors whose parameters are to be extracted. After entering your DUTs for BSIM3 modeling, use the *Display* button under *Size Category* on the left side of this folder to get a graph of the actual device geometries. See the following figure (applicable only for BSIM3).

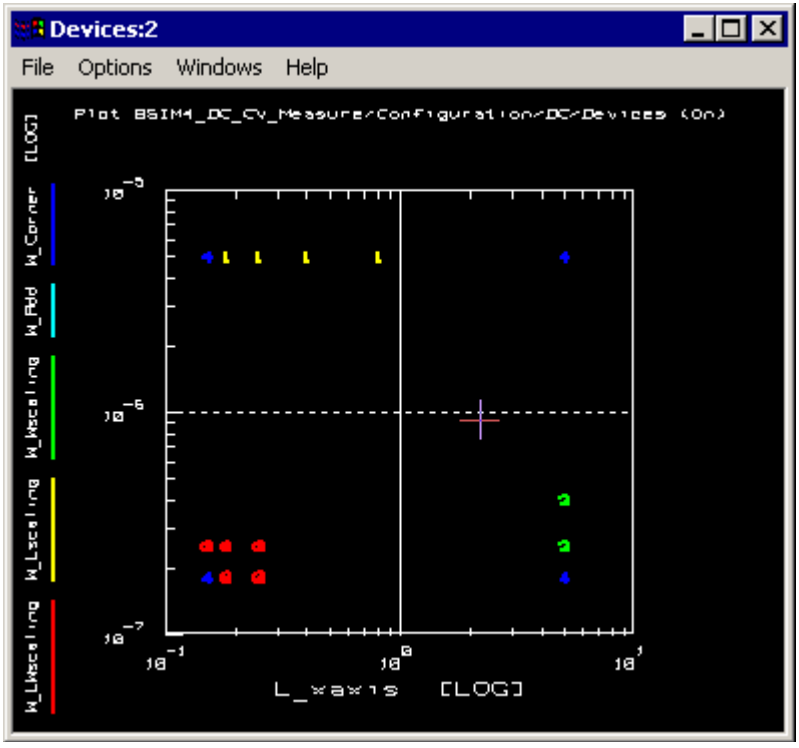


Figure 19 Display of device geometry distribution inside the BSIM3 GUI

You can use *Sort* to set the size category of your devices automatically. Otherwise, it is required to enter the size category manually using the form shown in Figure 21.

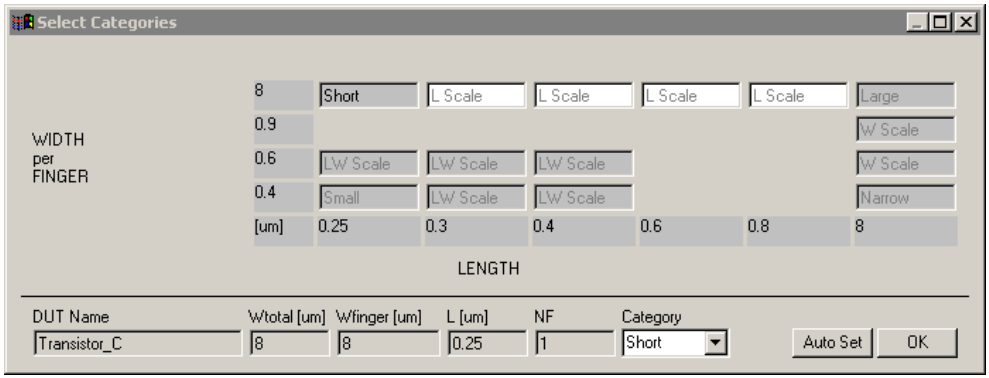


Figure 20 Set size category for BSIM3

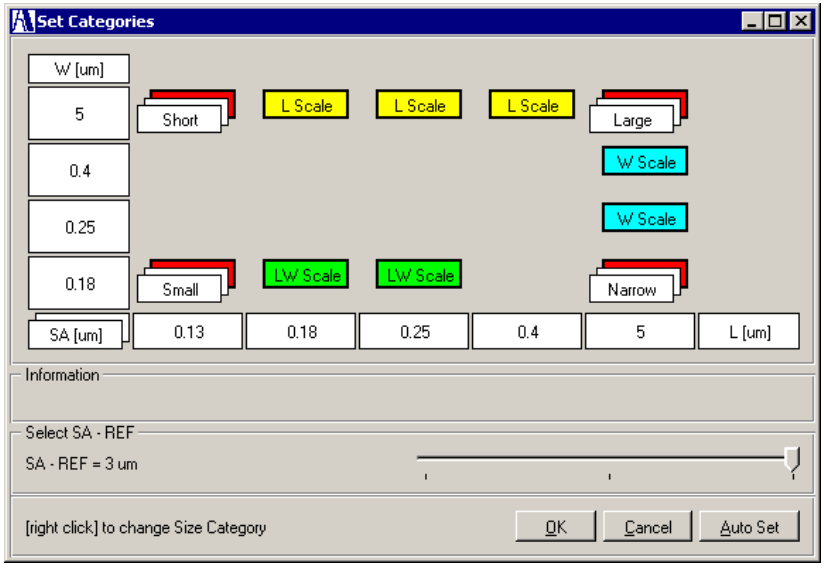


Figure 21 Set size category for BSIM4

The device category is used for extraction purposes. See "Transistors for DC measurements" on page 293 in the BSIM3 Characterization chapter for an explanation of categories and requirements for proper extraction of device

parameters as well as the paragraph about “Stress Effect Modeling” on page 372 inside the BSIM4 Characterization Chapter.

If you would like to delete DUTs:

- Choose *Delete* to the left of the folder. You will be prompted with a list of DUTs (Figure 22). Select the DUTs to be deleted and choose *Delete* on the *Delete DUT* form. A prompt dialog box appears. Select *OK* if you are satisfied with your choice of DUTs to be deleted.

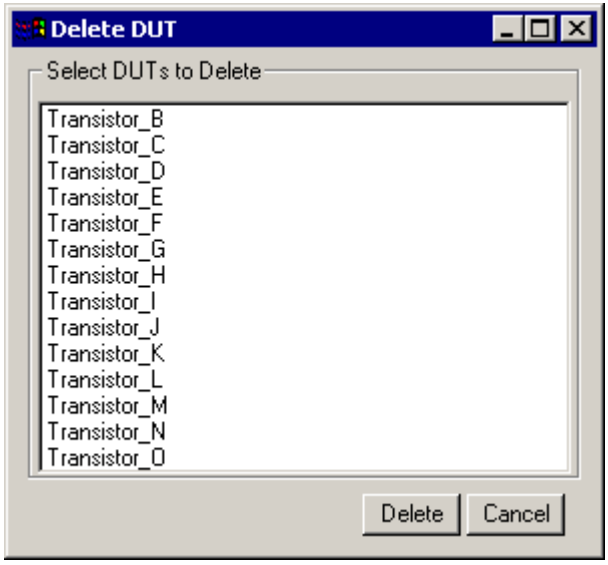


Figure 22 Delete DUTs form

According to your choice of temperatures on the *Temperature Setup* folder, one or more columns marked with the temperatures you’ve entered appear. The fields of those columns show either (0) for no measured data available, (M) for DUT already measured or (-) for DUT not to be measured at that temperature.

- To select devices to be measured at different temperatures: Choose *Temp Meas* to the left of the folder. You will be prompted with a list of DUTs. Select the devices to be measured at those temperatures entered in the Temperature Setup folder and click *OK*. You are able to select more than one DUT at a time for temperature measurement by repeated clicks on the ones to be chosen.

NOTE You cannot prevent a DUT from being measured at TNOM. All DUTs are to be measured at that temperature. If you have entered one or more temperatures on the Temperature Setup folder, the DUTs selected for temperature measurement are all measured at those temperatures. It is not possible to select a DUT for measurement at temperature T1 but not at another temperature T2.

NOTE To extract temperature effects on parameters, a large, a short, and a small device is necessary!

You can enter a comment for each DUT. If you are using a switch matrix, you are able to enter a module name and you have to enter the pin numbers of the switch matrix pin connections to the transistor in the fields below the node names (those fields are present only if the use of a switch matrix is selected on the *Switch Matrix* Tab). See Figure 23 for details.

NOTE For the Agilent E5250, the port number to be entered consists of 3 numbers.

If, for example, SMU1 shall be connected to Card No.1, Port No. 3: Enter the number 103 into the field below the transistors node name. Port No. 12 for Card No. 4 would have to be entered as 412.

NOTE

When using module names to measure devices with probe cards, pay attention to the node numbers you are entering. Each device uses 4 connections to the switch matrix. You have to enter the correct pin numbers for each DUT and must not exceed the total pin count for each port of your matrix.

Connections to the DUTs

The following figure shows an example for a connected device under test (DUT) to the source measurement units (SMUs) during DC Transistor measurements. The node numbers shown are to be entered into the fields below the terminal names on the DC Transistor DUTs folder, see Figure 14 on page 93. The above mentioned figure shows “0” under all terminal names. Those numbers have to be changed to “10” for the gate, “12” for the source, “11” for the bulk, and “9” for the drain terminal to reflect the connections used inside the following schematic. Please be careful to use the numbers SMU1...SMU4 during hardware setup in IC-CAP, since these are required by the measurement module of the BSIM3/4 Modeling Packages.

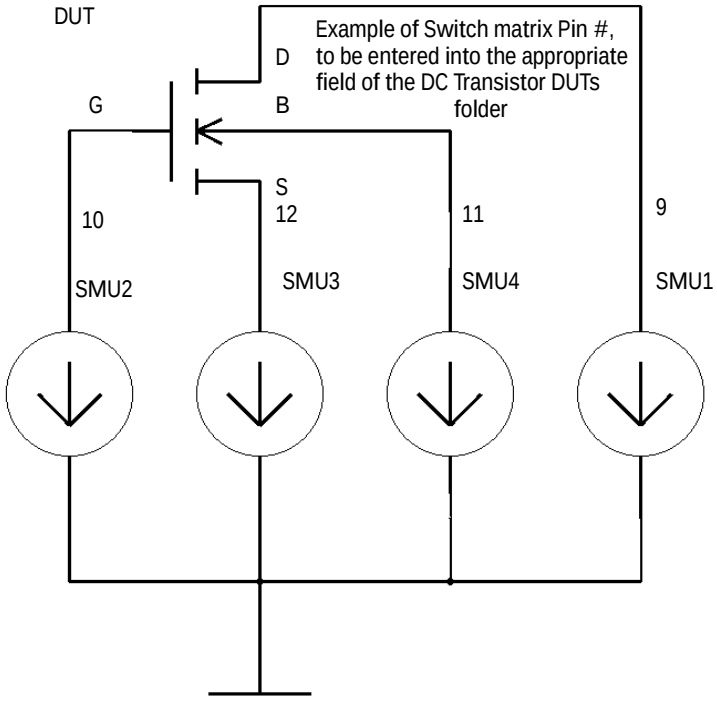


Figure 23 SMU connections to the device under test

Measurement of the devices

Once all DUTs are entered with their respective geometries, switch matrix pin connection and measurement temperatures, the actual measurement of devices can take place. You will find the appropriate buttons under the *Measurement* section on the left side of the *DC Transistor DUTs* folder.

- To start measurement of the devices: Click *Measure* and select the DUT or Module to be measured using the dialog box (Figure 24) that opens. You can select measurement temperature (if there is a temperature other than TNOM defined in the temperature setup folder) as well as a specific DUT or a Module (containing all DUTs to be measured at a specific temperature). If you select a temperature other than TNOM, (must be defined under *TemperatureSetup*), only the devices set up for measurement at that temperature are selectable for measurement. Start measurement with *Measure* (or *MeasureDUT* in BSIM4) on that dialog box. If measuring at elevated temperatures, be sure to wait until your devices are heated up or cooled down to the desired temperature.

NOTE

For your convenience, you will find a supplemental model file called "*prober_control.mdl*" which is suitable to be used with automatic temperature measurements under the directories *§ICCAP_ROOT/examples/modelFiles/mosfet/BSIM3(4)/examples/waferprober.* Tailor this model file to your specific Thermochuck model and requirements. Otherwise, be sure to set chuck temperature manually!

IC-CAP doesn't support heated chuck drivers.

If you select measurement of a module, all DUTs in this module are measured automatically if the use of a switch matrix is activated.

The DUTs/Setup folder in the *BSIM3/4_DC_CV_Measure* model contains an AutoMeasure setup for the Configuration DUT. Using this AutoMeasure setup, you can program automatic measurements for all DUTs in one module.

NOTE

Automatic measurement uses a macro for the wafer prober. This macro is programmed to start measurement as soon as the wafer prober has reached it's programmed destination.

You'll find the macro "Example_Wafer_Prober" and it's transforms together with a description of the transforms in the Macro folder of the *BSIM3/4_DC_CV_Measure* model.

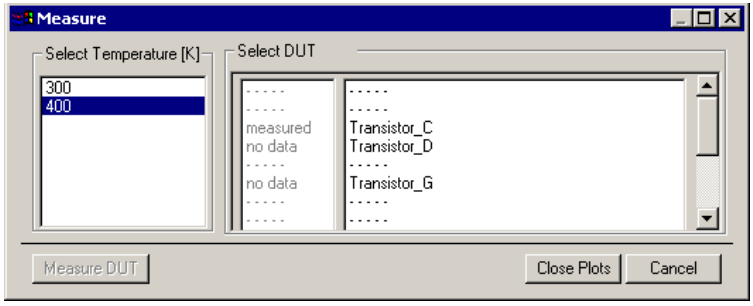


Figure 24 Select devices and measurement temperatures

- If you would like to clear some or all measured data, use *Clear*. You can select whether you would like to clear measured data of some or all DUT(s) at specified temperatures and choose *Clear Data* to delete measured data files.
- Using *Synthesize*, you are able to simulate data from existing model parameters. By selecting this feature, already measured data files are overwritten with synthesized data. You will be prompted (Figure 25) before existing data files are overwritten.

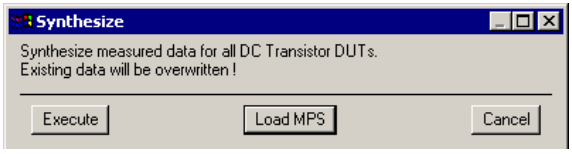


Figure 25 Prompt dialog for synthesizing measurement data

There is the choice of either synthesizing data or loading an MPS File.

This synthesized data uses the voltages set on the Measurement Conditions folder to generate “measurement” data from a known set of SPICE parameters. This feature might be especially useful to convert parameters of other models into BSIM4 parameters by loading the created “measurement data” into the extraction routines and extract BSIM4 parameters.

- For a glance at the diagrams that have just been measured, choose *Display Plots*. You will see a dialog box (Figure 26) to select which measured data set to be displayed. After choosing the plots, choose *Display Plots* on that dialog box to open the selected plots. This is a convenient way to detect measurement errors before starting the extraction routines.

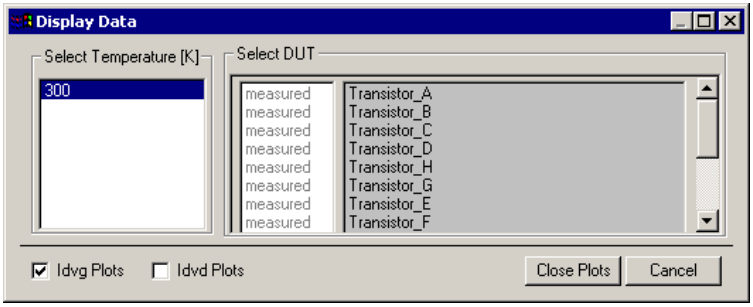
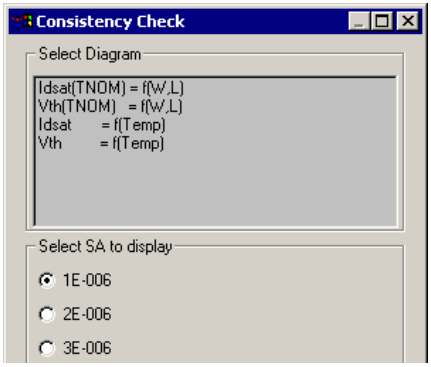
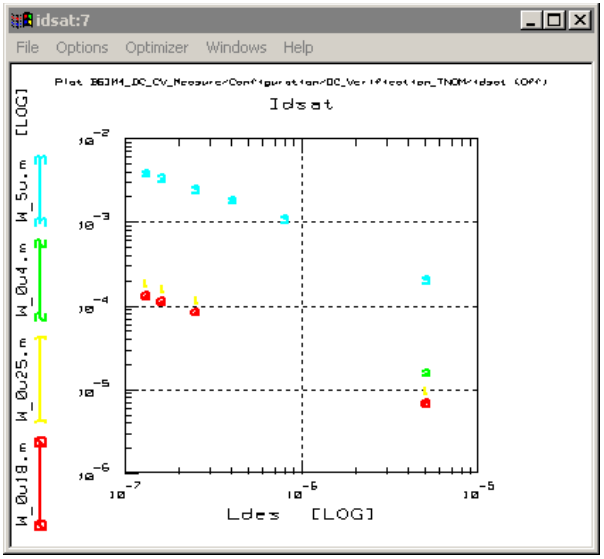


Figure 26 Selecting measured data to be displayed

- If you are satisfied with the data plots you’ve just measured, choose *Close Plots* to close the displayed plots of measured data, either on the **DisplayData** window or on the GUI window.
- There is a *Data consistency* check available to see if measurement errors have occurred. Click *Display Plots* to choose the desired type of plot out of a list of available plots for a quick consistency check.



- Select one of the available plots to see a normalized display of the saturation current of all measured devices, for example. The plot to be displayed looks similar to the one shown below. For an explanation of this consistency check feature, see "Consistency Check of DC measurement data for multiple measured devices" on page 234.



Capacitance DUTs

This folder is used to measure capacitances of devices and to display measured data.

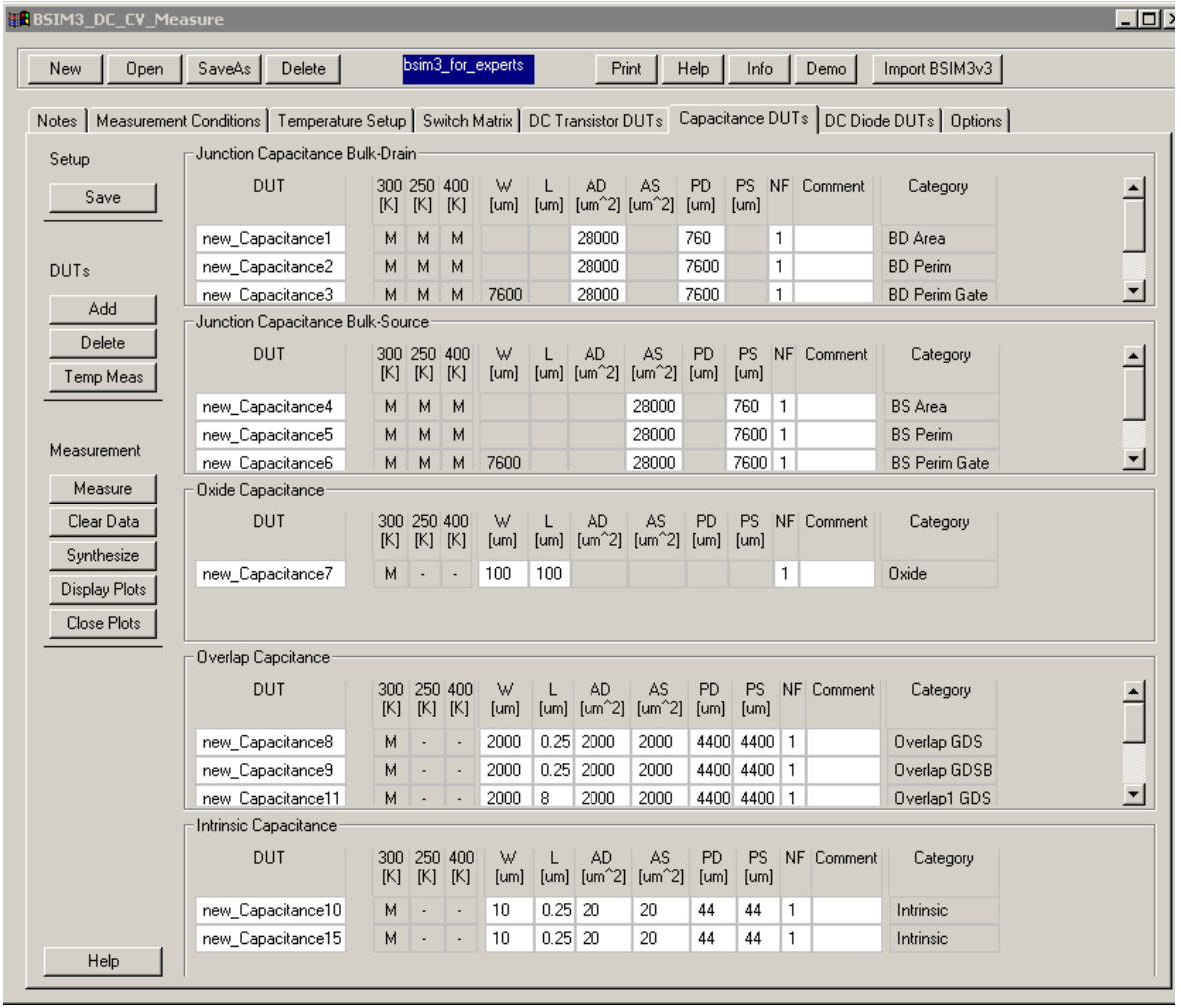


Figure 27 Capacitance DUTs folder

This folder provides fields to enter names of DUTs, geometries and switch matrix connections, and to select temperatures at which to measure the DUTs.

- To add new DUTs: Choose *Add* to the left of the folder. You will be prompted for a group of capacitances to add DUTs to (Figure 28). Select the desired group (junction bulk-drain or bulk-source, oxide, overlap, or intrinsic) and choose Add. New lines are added according to the selection you made.

NOTE

Selecting overlap capacitances actually adds two DUTs: Overlap_GDS and Overlap_GDSB. It is required for proper parameter extraction to measure both DUTs and extract the parameters from both measurements. Therefore, it only makes sense to add those DUTs together.

Since oxide capacitance requires only one test structure, you are able to have only one oxide capacitance DUT.

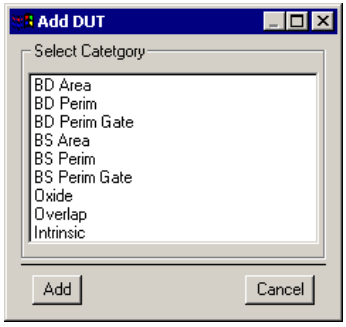


Figure 28 Add DUTs to measure capacity

For each line, you can change the predefined name for the DUT and enter necessary geometrical data. For your convenience, only relevant data should be entered for a specific group of capacities. Relevant data fields are shown with white background and can be edited. Gray shaded data fields are not editable. For example, DUTs to measure bulk-drain junction capacitances do not require gate length and width (L, W), source area (AS) and perimeter length of

source (PS) geometrical data. You only have to provide drain area (AD) and drain perimeter (PD) as well as the number of device fingers (NF) of the transistor to be measured. See Figure 29 for some details on capacitances and geometries.

NOTE W, AD, AS, PD, and PS are total values including all fingers of the device!

NOTE Usually, you use single finger transistors for DC measurements. Multifinger devices are common only in high frequency characterization of MOS devices, since the input resistance of a network analyzer is typically 50 Ohm.

Remember, all geometries are to be given in microns (µm).

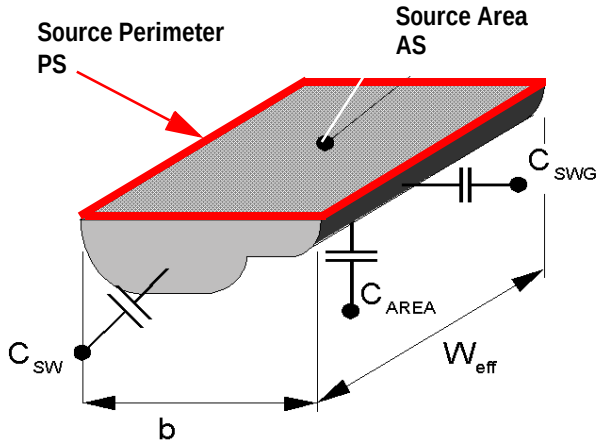


Figure 29 Device geometries

According to your choice of temperatures on the *Temperature Setup* folder, one or more columns marked with the temperatures you've entered appear. The fields of

those columns show either (0) for no measured data available, (M) for DUT already measured or (-) for DUT not to be measured at that temperature.

You can enter a comment for each DUT. When using a switch matrix for capacitance measurements, you are able to enter a module name to measure one complete module with all its DUTs at once. This is intended for use with a prober card and taking measurements using the “step and repeat” function of a wafer prober. If you are using a switch matrix, you must enter the pin numbers of the switch matrix pin connections to the capacity you‘re about to measure. The fields for high and low connection of the CV measuring instrument is marked H or L respectively. See “Test Structures for CV Measurements” on page 296 for details on device geometries and requirements for proper extraction of capacitances of your devices like test lead connections and so on.

- To delete DUTs: Choose *Delete* to the left of the folder. You will be prompted with a list of DUTs. Select the DUT(s) to be deleted and choose Delete on the *Delete DUT* folder. A prompt dialog box appears. Select *OK* if you are satisfied with your choice of DUTs to be deleted.
- To select devices to be measured at different temperatures: Choose *Temp Meas* on the left side of the folder. You will be prompted with a list of DUTs. Select the devices to be measured at those temperatures entered in the *Temperature Setup* folder and click *OK*.

NOTE

You cannot prevent a DUT from being measured at TNOM. All DUTs are measured automatically at that temperature. If you have entered one or more temperatures on the Temperature Setup folder, the DUTs selected for temperature measurement are all measured at those temperatures. It is not possible to select a DUT for measurement at temperature T1 but not at another temperature T2.

- To start measurement of the devices: Choose *Measure* and select the DUT(s) to be measured on the dialog box that opens. You can select measurement temperature (if there is a temperature other than TNOM defined in the temperature setup folder) as well as a specific DUT or all DUTs. Start the measurement with *Measure* on that dialog box. If measuring at elevated temperatures, be sure to wait until your devices are heated up or cooled down to the desired temperature.

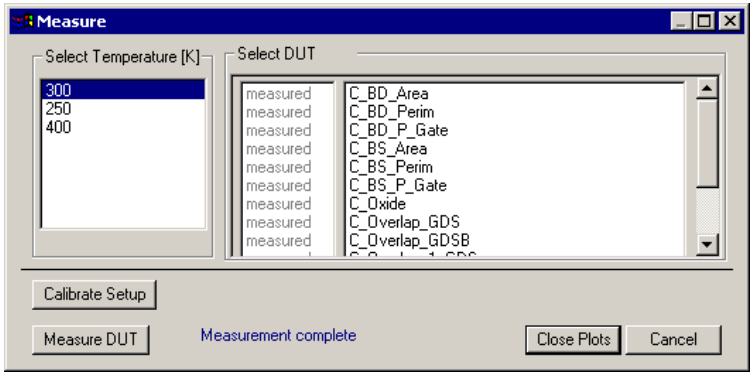


Figure 30 Selecting DUTs to be measured

- If you would like to clear some or all measured data, choose *Clear*. You can select whether you would like to clear measured data of some or all DUT(s) at specified temperatures and click *Clear Data* to delete measured data files.

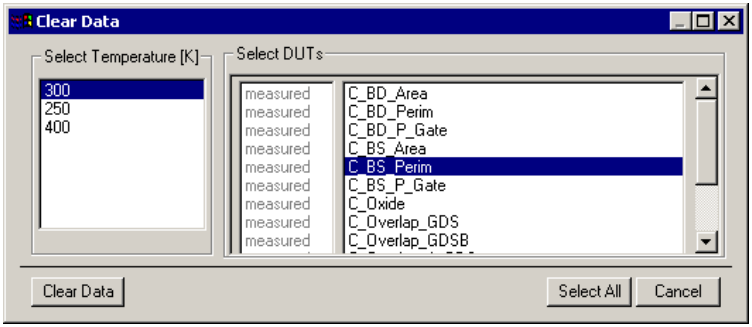


Figure 31 Selecting capacitance DUTs to be deleted

- Using *Synthesize*, you can simulate capacitance data from existing parameters. This synthesized data uses the voltages set on the *Measurement Conditions* folder to generate “measurement” data from a known set of SPICE parameters. This might be especially useful to convert parameters of other models into BSIM3 or BSIM4 parameters by loading the created “measurement data” into the extraction routines and extract parameters for the desired model.

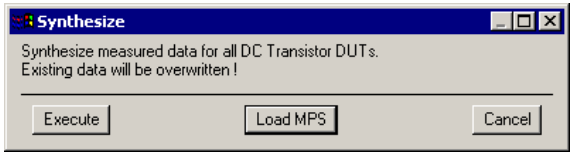


Figure 32 Synthesize Data Prompt

- For a glance at the diagrams that have been measured, click *Display Plots*. You will see a dialog box to select which measured data set you would like to display. After choosing the plots you would like to see, click *Display Plots* on that dialog box to open the selected plots. This is a convenient way to detect measurement errors before starting the extraction routines.

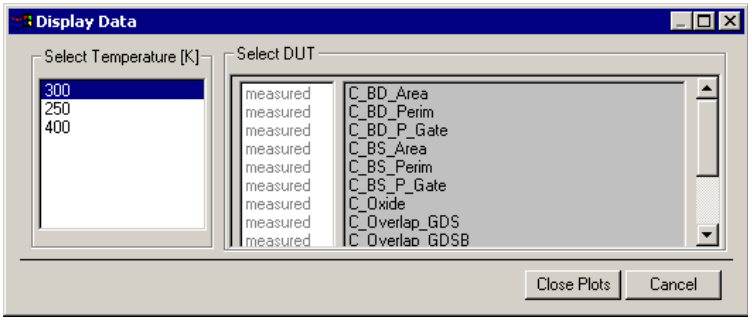


Figure 33 Select plots to be displayed

- Choose the plots of measured data to be displayed.
- If you are satisfied with the data you have just measured, choose *Close Plots* to close the windows that show diagrams of measured data.

Physically connecting Test Structures to Your Capacitance Measurement Device

Figure 34 shows how to connect the CV instrument to measure oxide and overlap capacitances. See also the paragraph on test structures for CV measurement. In Table 21 on page 296 you'll find recommended test structures for specific capacitances to be measured together with recommended instrument connections.

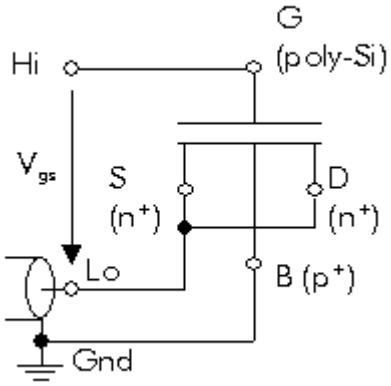


Figure 34 Measurement of oxide and overlap capacitance

The following figure shows a typical gate-to-drain/source overlap capacitance diagram that you would expect to measure with this type of connection and the default values for *Start*, *Step*, and *Stop* voltage V_G.

NOTE

To correctly extract overlap capacitance effects, two devices are essential: Standard CV measurement masks the channel capacity in short channel devices. This is the so called Short Channel Effect.

To overcome this masking, you need a short channel device for proper extraction of overlap capacitance parameters. To extract the parameter NGATE, you need to measure a long channel device in inversion since there is no short channel effect present in such a device.

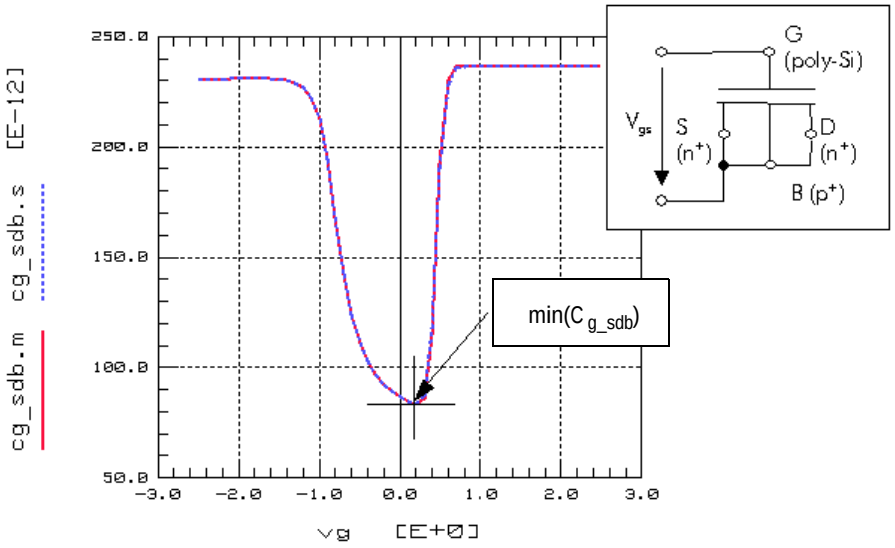


Figure 35 Example diagram of measured overlap capacity

Test Structures for CV Measurements

See Table 21 on page 296 for a table of recommended test structures for CV measurements.

DC Diode DUTs

This folder provides fields to enter names of DUTs, geometries and switch matrix connections, and to select temperatures at which to measure the DUTs. Don't forget to *Save* your setup after you enter the DUT data. Table 21 on page 296 briefly describes usable test structures to characterize diode behavior.

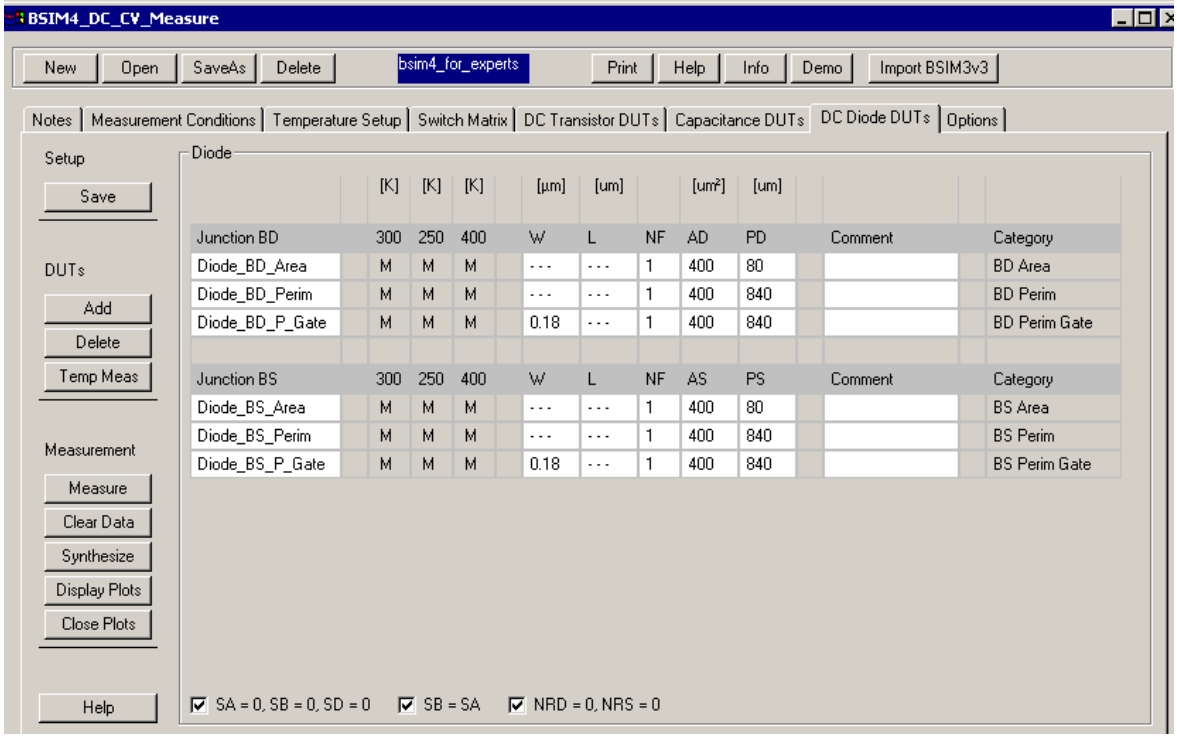


Figure 36 DC Diode DUTs folder

- To add new DUTs: Choose *Add* on the left side of the folder. You will be prompted with a list to select DUTs to add. Select the desired DUT(s) and click *Add*. New lines are added according to the selection you've made.

NOTE If you have entered all necessary categories, clicking *Add* will not open a window to select new diode DUTs, since all are present. Measuring more diode DUTs will not create new information, since the measured values will be the same as the one's that have been measured already.

For each line, enter a name for the DUT and necessary geometrical data. For your convenience, only relevant data is to be entered for specific diodes. Relevant data fields show a

white background, irrelevant data fields show a dashed line. For example, DUTs to measure bulk-drain diodes do not require source area (AS) and perimeter length of source (PS) geometrical data. You only have to enter drain area (AD) and drain perimeter (PD) as well as the number of device fingers (NF) of the diode to be measured. Remember, all geometries are to be given in microns (µm).

NOTE W, AD, AS, PD, and PS are total values including all fingers of the device!

According to your choice of temperatures on the *Temperature Setup* folder, one or more columns marked with the temperatures you've entered appear. The fields of those columns show either (0) for no measured data available, (M) for DUT already measured or (-) for DUT not to be measured at that temperature.

- You can enter a comment for each DUT. If you are using a switch matrix, you can enter a module name as well as the pin numbers of the switch matrix pin connections to the transistor. Only relevant connections are to be entered, in case of the bulk-drain diode, no source connection must be entered (the appropriate field shows a dashed line). See Figure 29 on page 110 for details on device geometries and Table 20 on page 296 for requirements on a proper extraction of diode data.
- To delete DUTs: Choose *Delete* to the left of the folder. You will be prompted with a list of DUTs. Select the DUT(s) to be deleted and click *Delete* on the *Delete DUT* folder. A prompt dialog box appears. Choose *OK* if you are satisfied with your choice of DUTs to be deleted.
- To select devices to be measured at different temperatures: Choose *Temp Meas* on the left side of the folder. You will be prompted with a list of DUTs (see the following figure). Select the devices to be measured at the temperatures entered in the Temperature Setup folder and click *OK*.

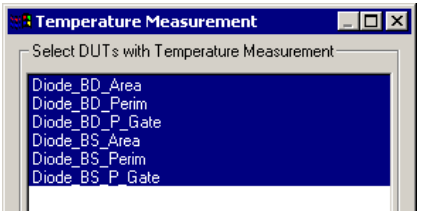


Figure 37 Dialog box to define DUTs to be measured at temperatures entered on the Temperature Setup folder

NOTE

You cannot prevent a DUT from being measured at TNOM. All DUTs are measured automatically at that temperature. If you have entered one or more temperatures on the Temperature Setup folder, the DUTs selected for temperature measurement are all measured at those temperatures. It is not possible to select a DUT for measurement at temperature T1 but not at another temperature T2.

- To start measurement of the devices: Click *Measure* and select the DUT(s) to be measured on the dialog box that opens (Figure 38). You can select measurement temperature (if there is a temperature other than TNOM defined in the temperature setup folder) as well as a specific DUT. Start the measurement with *Measure* on that dialog box. If measuring at elevated temperatures, be sure to wait until your devices are heated or cooled down to the desired temperature.

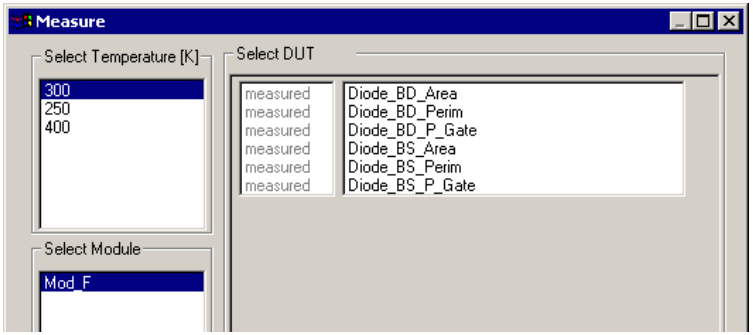


Figure 38 Measurement selection window

- If you would like to clear data of some or all measured DUTs, use *Clear*. Select whether you would like to clear measured data of some or all DUT(s) at specified temperatures, (see Figure 39) and click *Clear Data* to delete measured data files.

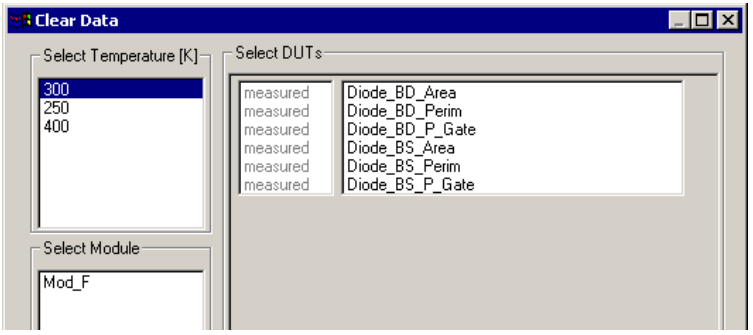


Figure 39 Clear measured data

- Using *Synthesize*, you can simulate data from existing parameters. This synthesized data uses the voltages set on the *Measurement Conditions* folder to generate “measurement” data from a known set of SPICE parameters.

- For a glance at diagrams that have just been measured, use *Display Plots*. You will see a dialog box to select which measured data set you would like to display. Choosing the plots you would like to see, opens the selected plots. This is a convenient way to detect measurement errors before starting the extraction routines.

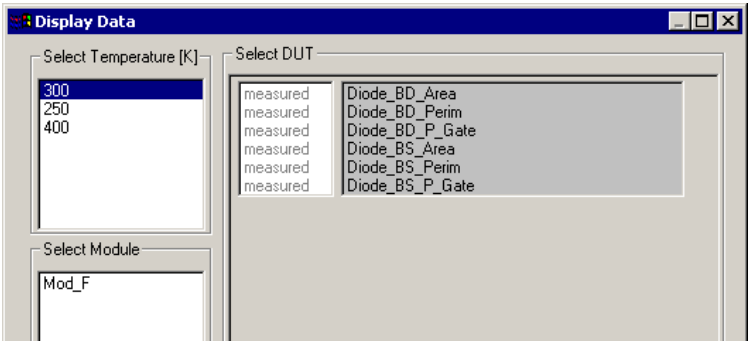


Figure 40 Display plots dialog box

- If you are satisfied with the data you just measured, use *Close Plots* to close the windows that show diagrams of measured data.

Drain / Source – Bulk Diodes for DC Measurements

For test structures to measure DC Drain/Source-to-Bulk diodes, see "Drain/Source – Bulk Diodes for DC Measurements" on page 121.

Options

This folder lets you define options for the appearance of the plot windows.

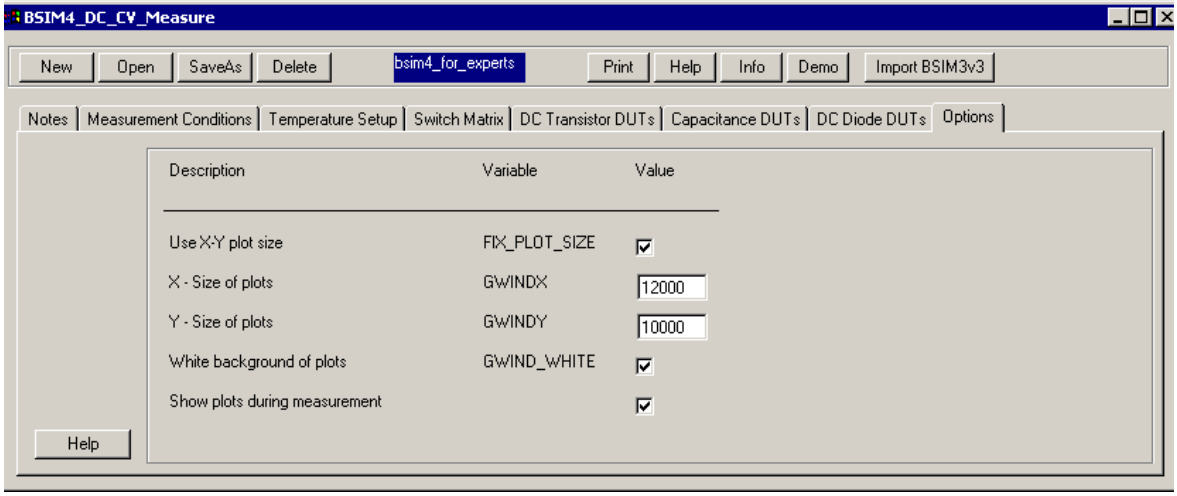


Figure 41 Options folder

To change the plot window size, deselect the *FIX_PLOT_SIZE* check box and enter the desired X- and Y-size into the respective fields. You are able to change the background color of the plot window from black to white by activating the *White background of plots* field, *GWIND_WHITE*. There is a check box “Show Plots during measurements,” which when activated, enables the display of the measured data.

RF Measurement

This section provides information on RF measurements using the BSIM3/BSIM4 Modeling Package. Starting the RF module opens a GUI divided into a number of folders for each task. The top row buttons are the same as in the DC and CV measurement module. Assuming that you are already familiar with the functions of these buttons, we will not describe their purpose. You can easily check their use in "DC and CV Measurement of MOSFET's for the BSIM3 and BSIM4 Models" on page 77.

RF Measurement Notes

The RF measurement module also contains a *Notes* folder to take notes on the project. It has the same look as the Notes folder of the DC/CV-measurement module, see "Project Notes" on page 80.

RF Measurement Conditions

The first task during RF modeling is to set up measurement conditions. Use the *Measurement Conditions* folder shown in the following figure to enter the measurement conditions at which you would like to measure and extract RF parameters.

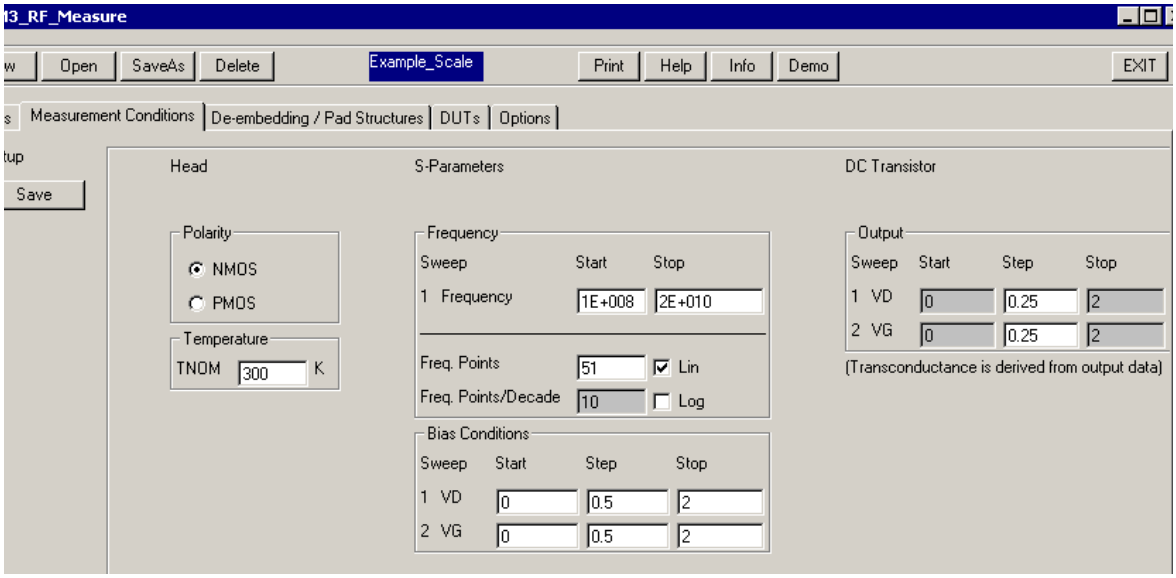


Figure 42 Measurement Conditions folder

Select the *Polarity* of the transistor to be measured (NMOS or PMOS) using the appropriate check box. Enter the measurement temperature TNOM, if the measurement is being conducted at any other temperature than the default of 300K.

NOTE Be sure to enter the temperature value in Kelvin!

The *DC Transistor* fields in this folder allow you to set sweep values for gate and drain voltages respectively. Enter *Step* values for the Output fields. For an in-depth description, see "DC Measurement Conditions" on page 81.

The purpose of this field is to define measurement of DC characteristics of multifinger transistors used for RF NWA measurements. This step is necessary, since the DC behavior of a multifinger transistor differs from that of a single finger

transistor. During DC measurement and extraction, a single finger transistor is being used, whereas a multifinger transistor is used in RF measurements to deliver sufficient drain currents for network analyzers to improve the measurement accuracy. Actual transistor DC measurements are used to set start points for S-parameters at low frequencies and control extraction at those points.

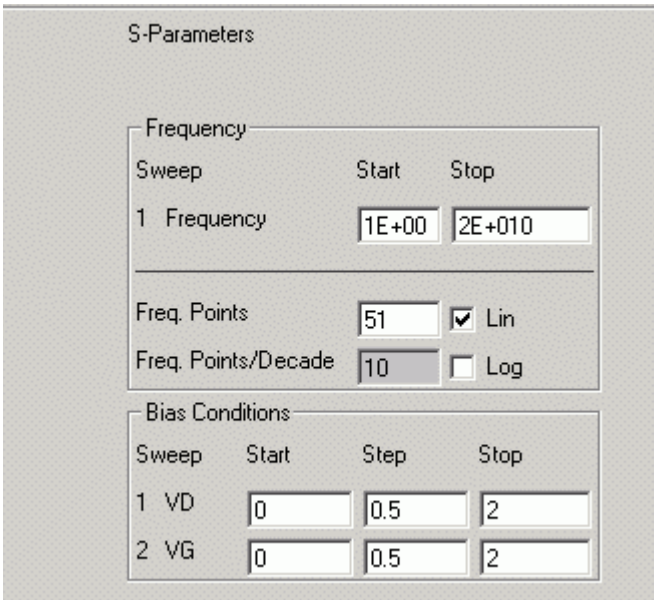


Figure 43 S-parameter part of the Measurement Conditions folder

The *Measurement Conditions* folder provides fields to enter conditions for S-Parameter measurements. Enter *Start* and *Stop* frequency, choose the desired sweep (*Lin*ear or *Log*arithmic) and enter the number of Frequency Points to be measured during linear sweep or the number of Frequency Points/Decade for logarithmic sweep. Only the field ahead of the chosen sweep type (Lin or Log) is enabled allowing data to be entered.

Use the field *Bias Conditions* to enter sweep voltage *Start*, *Step*, and *Stop* values for drain and gate voltages during S-Parameter measurements.

NOTE Be careful not to exceed the maximum DC Input voltage of the Network Analyzer used during measurements!

Deembedding / Pad Structures

The Deembedding/Pad Structures folder enables you to select the type of pad structure used for deembedding the parasitics of the measurements. Measurement of the transistors for parameter extraction requires connecting the devices to the instruments. Therefore, the basic transistor element to be measured must be connected using pads and metal connections on the wafer. In order to get the parameters of the basic transistor without metal connections and pads, the parasitics must be deembedded from measurement results. A device library should contain only the basic transistor element. The connections to other elements in a circuit have to be modeled separately since this is part of the interconnection between elements on a chip.

Basically, you perform error correction of your network analyzer in order to eliminate measurement errors resulting from cable connections used to interface the analyzer to the wafer prober and up to the probe tips. Your test chip design must contain structures to eliminate the parasitics as a result of connecting prober needles via metal lines to transistor terminals.

This folder is intended to define the structures used to deembed the transistor parameters from measured ones.

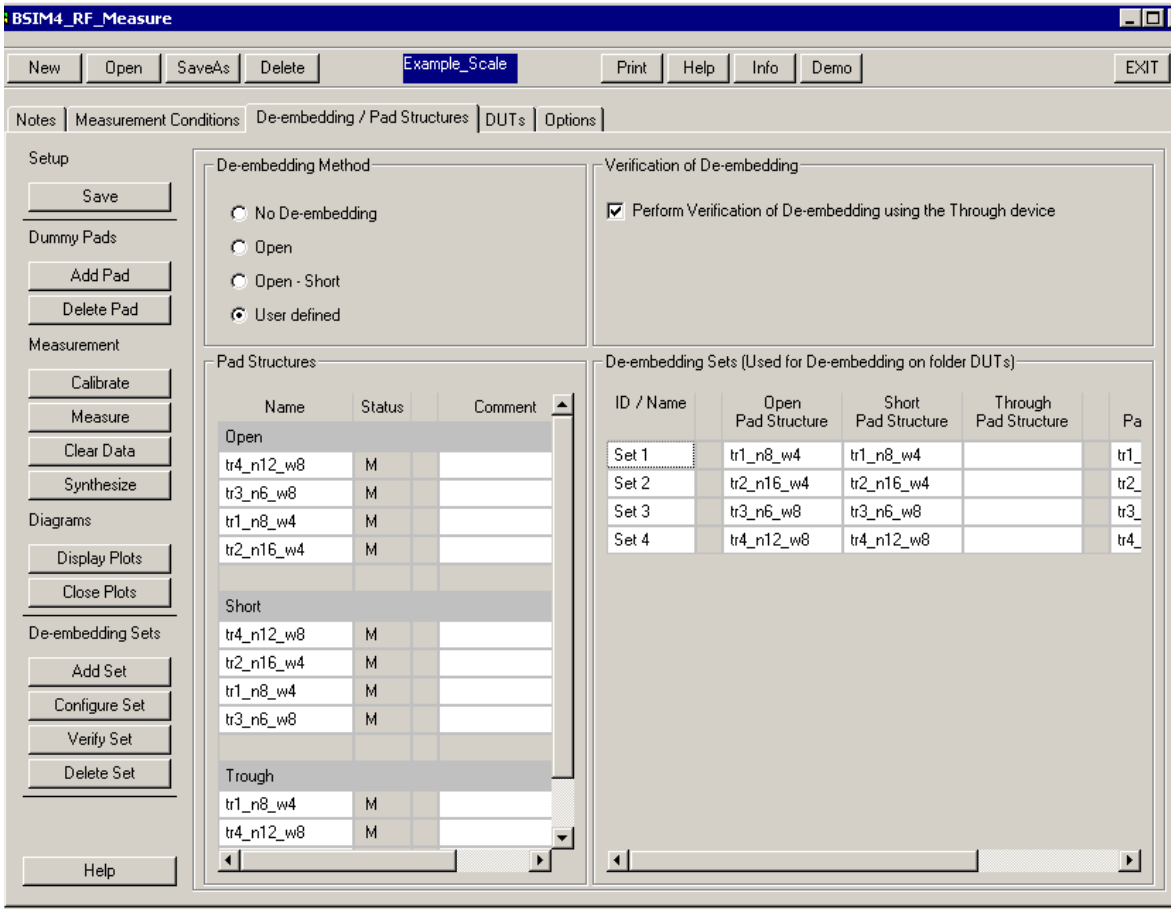


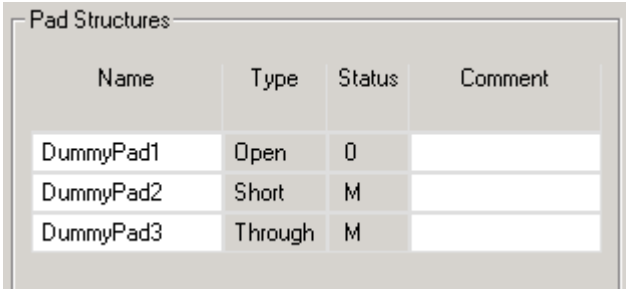
Figure 44 Deembedding pad parasitics

The section *De-embedding Method* provides check boxes to select the method for de-embedding to be used. Check one of *No De-embedding*, *Open*, *Open-Short*, or *User defined*. Your selection of the de-embedding method will affect the definition of de-embedding sets described later in this paragraph.

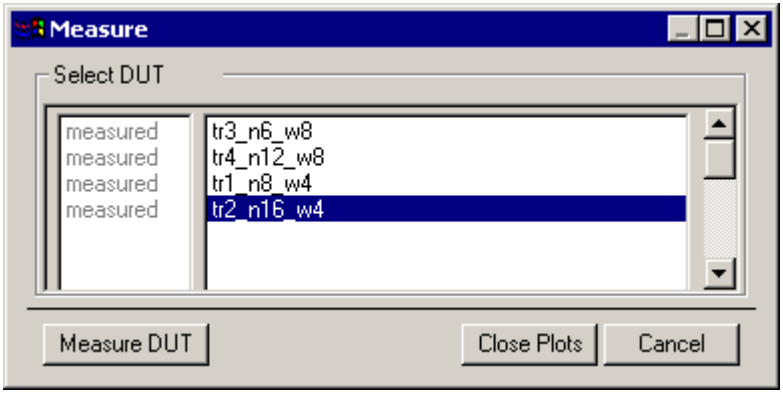
There is a section *Verification of De-embedding* where you can check a box to perform verification of de-embedding using the through device, if applicable.

The field *Pad Structures* is intended to declare dummy pads for de-embedding purposes. Click Add Pads and select the type of pad by clicking *Open*, *Short*, or *Through* on the appearing window, see below.

A new line will be inserted inside the field *Pad Structures*. You can change the name of the dummy pads as you like. The following rows are showing the type of deembedding pad (open, short, or through), status (not measured, showing “0”, measured, showing “M” or not applicable, showing “-”) as well as a user comment.

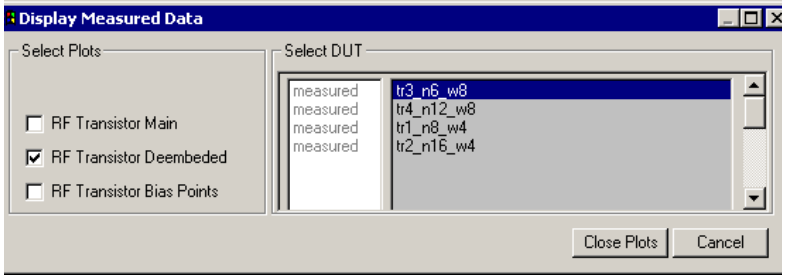


The command buttons under *Measurement* are used to perform the measurement of the defined pads. Press *Measure* and select the pad you would like to measure at the appearing *Measure* form before clicking *Measure*.



After the measurement has been performed, plots will be displayed so that the measured results could be checked for plausibility of the measured data. Click *Close Plots* to close the displayed plots of measured pads.

It is possible to view the measured pad data at any time after measurement has been performed by clicking *Display Plots* under the *Diagrams* part of the command buttons menu and select the pad to be displayed, see screen shot below. The data will be displayed on Smith charts and can be closed using the *Close Plots* button.

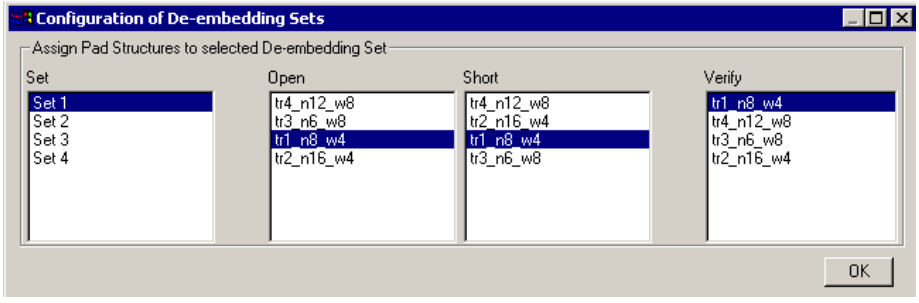


If you have chosen to perform a *Verification of de-embedding* by activating the *Perform Verification of de-embedding Using...* check box on the top right part of the folder, the field *Deembedding Sets* is being activated and you are able to configure the sets for use within the *DUTs* folder. Save the settings before configuring de-embedding sets.

A de-embedding set actually is a combination of pad structures to be use for de-embedding of measured devices. They can be used for one or more devices and can consist of any available pad structure.

Click the *Add Set* button once for each set to be used. A line for each set will be added. You are able to overwrite the predefined name of the sets.

Click *Configure Set* to assign a pad structure to a selected de-embedding set on the *Configuration of de-embedding Sets* form. Depending on the type of de-embedding you have chosen (*No de-embedding*, *Open*, *Short* or *User Defined*) in the *De-embedding Method* part of this folder, you are able to assign the respective pads to the sets. In other words, if you select *Open* as the de-embedding method, you can passing only pads of type *Open* to selected de-embedding sets. If you select *Open-Short* or *User defined*, you are able to assign pads of type *Open*, *Short* and *Through* to a set.



You can select the check box *Perform Verification of deembedding using the through device* if you have a Through device available on your test chip. Only after activating the verification, you can click *Verify Set* to perform the verification of a selected set.

NOTE

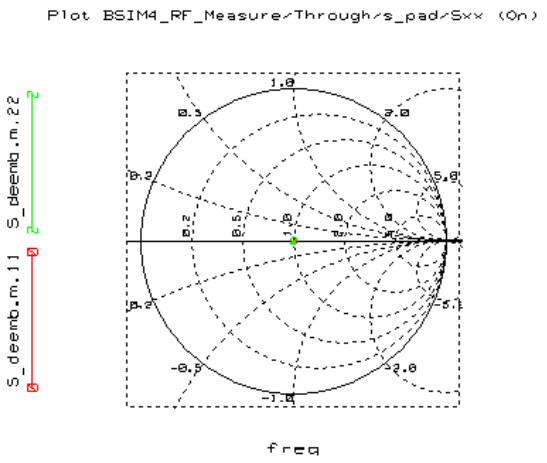
Ideal de-embedding means:

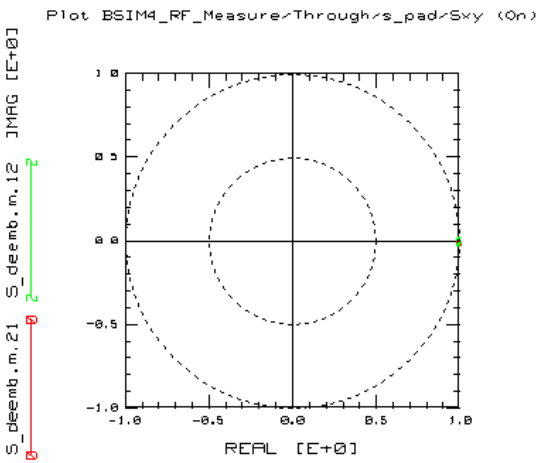
The S-Parameters should behave like an ideal matched transmission line with $Z_0 = 50\Omega$ and a time delay T_D representing the electrical length of the TROUGH device measured.

S11 and S22 should be concentrated at the center of the Smith chart, while S21 and S12 both start at $(1 + j0)$ and turn clockwise on the unity circle. If this is not true, the following items should be checked:

- Is the calibration OK?
- If the OPEN method is used: De-embedding quality can be enhanced by switching to the OPEN-SHORT method.
- For very high frequencies (approximately above 30 GHz) the assumptions for using the OPEN-SHORT method might not be given. You should probably change to an alternate calibration method.

The verification will be done and the plots will be displayed after clicking *OK* on the upcoming message window. The following plots show what is to be expected for correct deembedding.





An error message will show up if one of the sets is not configured correctly.

After the de-embedding is done, you can assign the appropriate pad sets on the *DUTs* folder to their respective devices.

Deembedding of parasitic structures

The section "Test Structures for S-parameter Measurements" on page 299 describes the effects of deembedding. It is intended to give you an insight into deembedding methods and describes the results of S-Parameter measurements with and without deembedding.

DUTs

The DUTs folder is used to define transistor geometries for the DUT to be measured. Following the column for entering the name of the DUT, there is a column showing the status of the DUT. This column shows "0" if no measurement and de-embedding has been performed. It changes to "M" if a measurement has been performed and to "M, D" if measurement and de-embedding has been done. The geometries to be entered into the following columns are **L**ength and **W**idth of the transistor, **D**rain and **S**ource **A**rea and **P**erimeter Length (AD, AS, PD, PS), **N**umber of transistor **F**ingers (NF) as well as the model selectors GEOMOD and RGEOMOD (applicable to BSIM4 only), which are set to their default values (0). The following columns NRS, NRD and MIN are originally BSIM4 parameters, but are used also inside the extended BSIM3 model from AdMOS.

NOTE W, AD, AS, PD, and PS are total values including all fingers of the device!

NOTE

The DUTs folder looks the same for BSIM3 and BSIM4. However, this folder contains some parameters and model flags that are only applicable for BSIM4. The default values for parameters not used in BSIM3 are set in such a way, that BSIM3 ignores them. This is done to make the form compatible for both models.

In BSIM4, the model selector GEOMOD is used to select a geometry-dependent parasitics model that specifies whether the end source/drain diffusions are connected or not. The default value is (0) – not connected. The parameter RGEOMOD is the source/drain diffusion resistance model selector. It specifies the type of end source/drain diffusion contact type: point, wide or merged contact. The default value is (0) – no source/drain diffusion resistance. See the BSIM4 manual from UC Berkeley [1] on page 11-5 and 11-6 for a definition of GEOMOD and RGEOMOD model selector values.

The parameters NRS, NRD and MIN are the layout dependent parameters **N**umber of **S**ource/**D**rain diffusion squares and **Min**imization of diffusion squares for even numbered devices. They are set to their default values (0), too.

The last column allows you to enter a comment for this DUT.

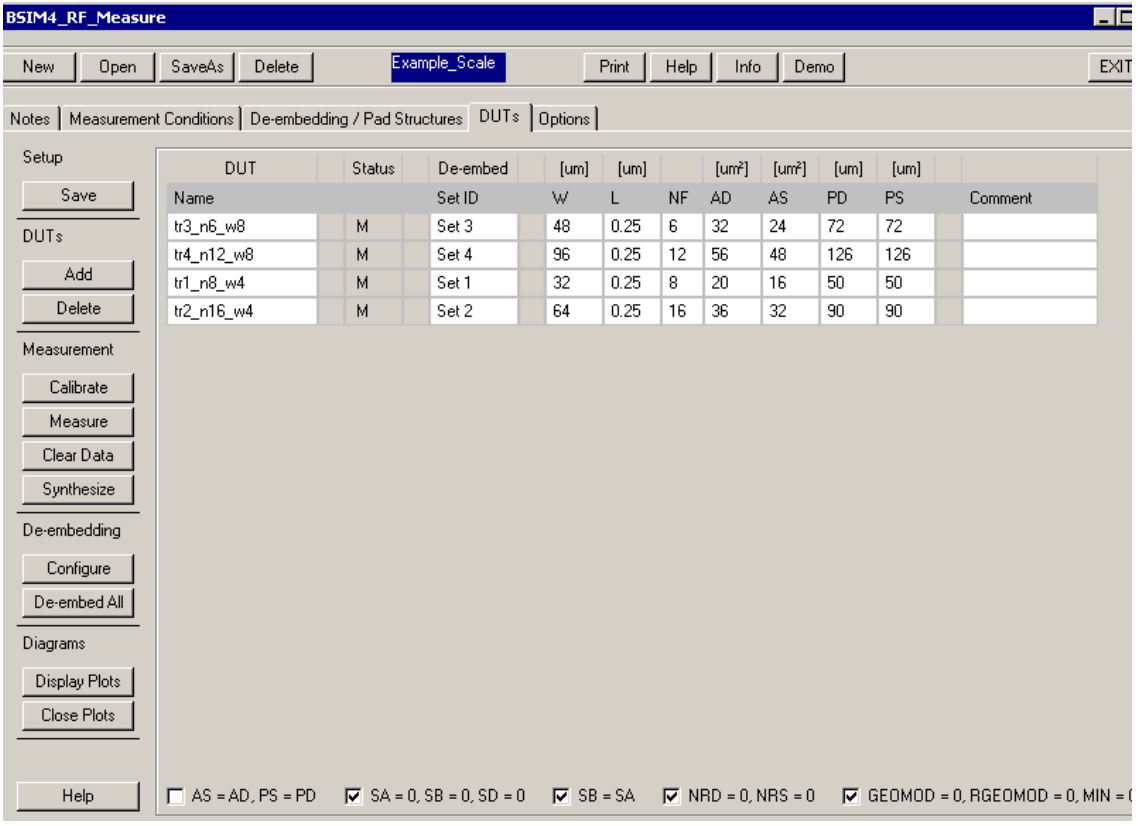


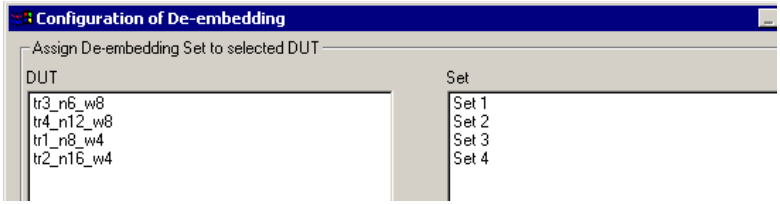
Figure 45 DUTs folder

The left side of the folder shows buttons to save the setup and to *Add* or *Delete* DUTs. There are also buttons associated with *Measurement*, *Deembedding*, and *Diagrams*.

The measurement buttons are used to *Calibrate* the network analyzer, to start a *Measure*ment using the *Start*, *Stop* and *Step* definitions on the measurement setup folder, to *Clear* measured data, or to *Synthesize* data. Once a DUT has been measured completely, the *Status* column will change from “0” to “M” to show the state of measurement.

Synthesize data performs a simulation of S-parameters, using the frequency definitions on the DUTs folder and a set of parameters loaded into the program from any other extraction task, to see correlations or to extract parameters into BSIM4 from another model release.

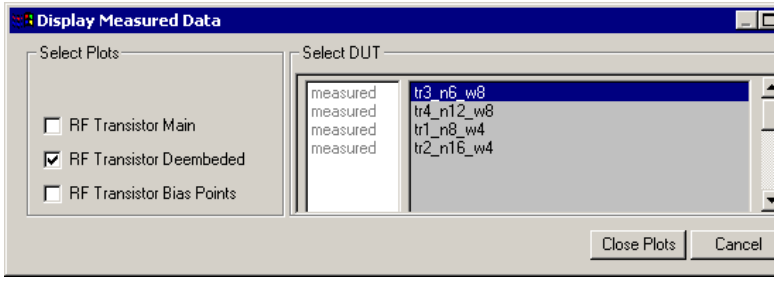
A *De-embedding* task starts with configuration before de-embedding the measurement setup (*De-embed All*). Click *Configure* to assign a de-embedding set defined on the *De-embedding/Pad Structures* folder to a specific DUT. You will get a list of DUTs and de-embedding sets defined for assignment.



Click the DUT, then click the SET to be assigned to the selected DUT.

Click *De-embed All* to start de-embedding measured data for each DUT with a de-embedding set assigned.

The Diagrams section is intended to check the measurement results in the form of diagrams, using the settings made on the Options folder. You can *Display* and *Close Plots* using the appropriate button for this task.



Check, whether you would like to see the plots of the raw data (*RF Transistor Main*, as measured), the de-embedded RF transistor data (*RF Transistor Deembedded*) and/or the bias points during measurement (*RF Transistor Bias Points*).

RF Measurement Options

Using this folder, you are able to set plot options like X and Y plot size and background color of the plot window. See "Options" on page 121 for details.

Extraction of DC/CV Parameters

Extraction of the complete BSIM3- or BSIM4-parameters is done using two different modules. There is a module inside each Modeling Package for extraction of DC/CV parameters and a module for RF parameters.

NOTE The RF extraction module needs start values for some parameters of a given process. Usually, those start values are taken from the DC/CV extraction process. Therefore, you should extract DC/CV parameters first.

The following figure shows the GUI used for extraction purposes. You can see the folders for the tasks during the extraction process. Again, ordered from left to right are folders for *Notes*, for measurement *Information*, and to *Initialize* the extraction process. The next folders are used for *Binning*, for setting up the *Plot Optimizer*, to *Extract* parameters from measurement data, to *Display* measured and simulated data to visualize the results of parameter extraction, to create a report in *HTML* format for publishing the parameters extracted together with some graphics of simulations using the parameters extracted, and to set some *Options* and *Boundaries* for the parameters to be extracted.

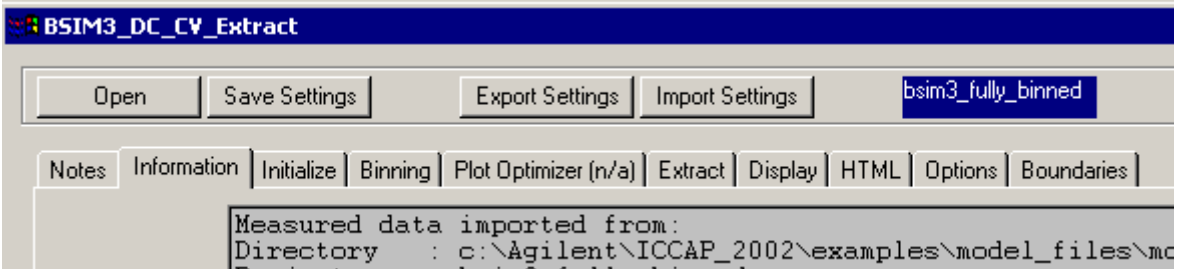


Figure 46 GUI for the Parameter Extraction Process

The top row of the GUI window contains buttons to perform file operations like *Open*, *Save Settings*, *Export Settings* and *Import Settings*. In the middle of the top row you'll find the

project name on blue background. To the right of the project name there is a *Help* button to open this manual. The *Info* button provides some information about the BSIM Modeling Package like version, date, and it´s creators—AdMOS.

The *Load Demo Data* button gives you the possibility to explore the features of the modeling packages without the need to measure data. For your convenience, examples are provided. If you press *Load Demo Data*, you will be prompted to copy and load the example files using the *Copy and Open Example* button on that form and filling in a path to a location where you have write access. This step is necessary, since the IC-CAP example directories are usually write protected and you need write access to modify the example files.

Far to the right you'll find the *Exit* button to leave the GUI.

If you would like to *Open* a project, you can select the project path and name in the *Open Project* dialog box. Using the *Export Settings* button opens the *Export Extraction Settings* dialog box which enables you to choose path and name of the saved extraction settings file. It is not possible to create a non-existing path on Windows. Instead, you must create the desired folders, if non-existent, using the Windows Explorer. The settings to be saved are the folders *Notes*, *Initialize*, *Extract*, and *HTML*.

NOTE

Opening a project takes some time. You are able to reduce this time by saving the complete ~.mdl file. Reloading the ~.mdl file is faster than opening a project. However, since the ~.mdl file contains a large amount of data which is already stored somewhere in the system, you need to have extra storage capacity on your hard disc.

You are able to *Import Extraction Settings* by selecting the path and name of the saved extraction settings file inside the *Import Extraction Settings* dialog box. This might be useful, for example, if you found a special extraction sequence that best fulfills the need of your parameter

extraction process. You save those sequence by exporting the settings to a file and using this file as template for following extraction processes.

NOTE Importing extraction settings will overwrite the actual settings within the active extraction process!

The tasks to be performed are ordered from the left to the right side of the IC-CAP/Main window. They should be performed in this manner. Some of the folders have default values for your convenience. If you are satisfied with the defaults, those folders could be left as they are. However, you are not required to follow this order.

DC Notes

There is a folder provided to take some Notes on the project. It has the same look as the one used in the Measurement modules—see Figure 3 on page 80, for example.

NOTE This folder is intended for notes on extraction. It will *not overwrite* your notes entered and saved during the measurement session.

DC Information

The next folder to the right gives you Information about the devices measured (see Figure 47). You'll find the type of MOSFET, measurement temperatures, DUT names together with their geometries and categories as well as the notes entered during the measurement. To the right of each DUT you can find a notice regarding the measurement status.

NOTE It is not possible to change the measurement information during the extraction session. This folder is for information on measurements only!

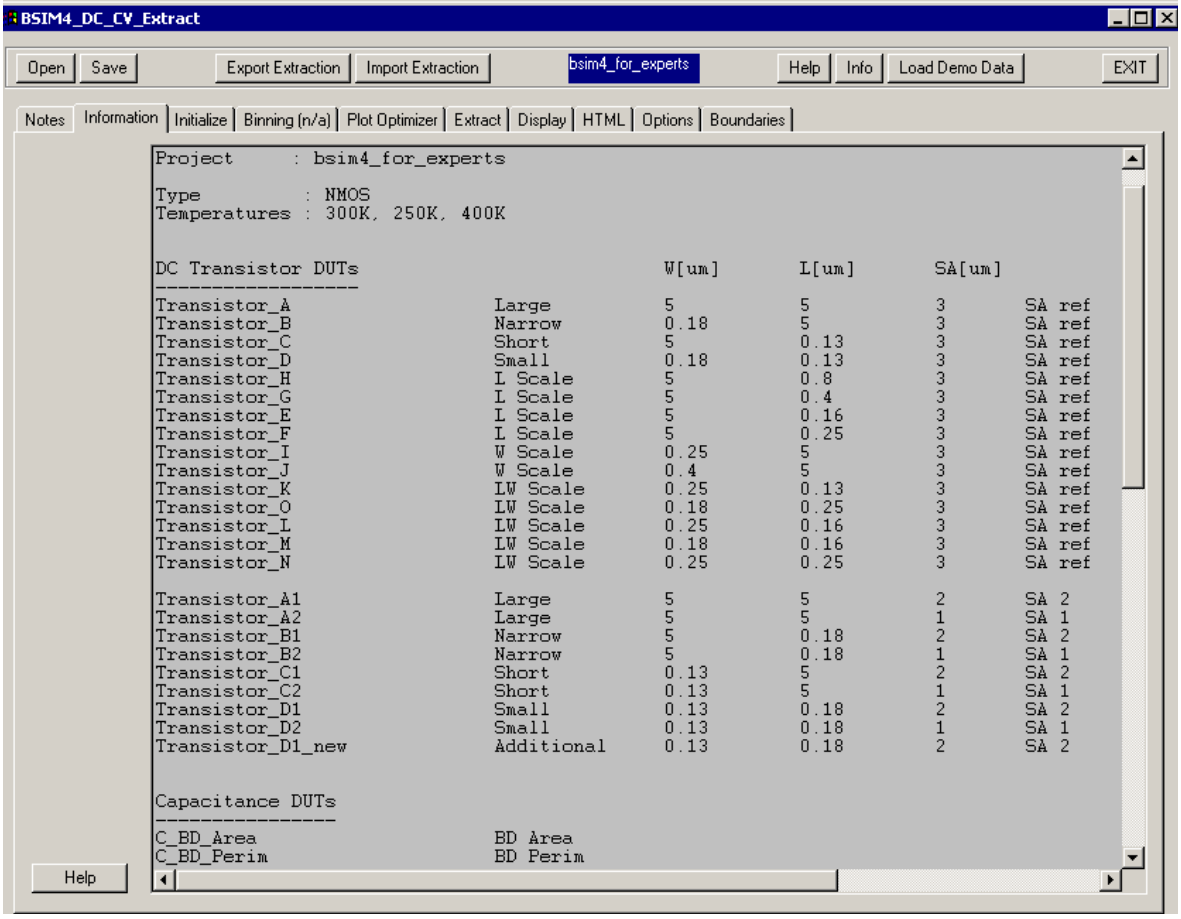

BSIM4_DC_CV_Extract
Open
Save
Export Extraction
Import Extraction
bsim4_for_experts
Help
Info
Load Demo Data
EXIT
Notes
Information
Initialize
Binning (n/a)
Plot Optimizer
Extract
Display
HTML
Options
Boundaries
Project : bsim4_for_experts
Type : NMOS
Temperatures : 300K, 250K, 400K
DC Transistor DUTs W[um] L[um] SA[um]
Transistor_A Large 5 5 3 SA ref
Transistor_B Narrow 0.18 5 3 SA ref
Transistor_C Short 5 0.13 3 SA ref
Transistor_D Small 0.18 0.13 3 SA ref
Transistor_H L Scale 5 0.8 3 SA ref
Transistor_G L Scale 5 0.4 3 SA ref
Transistor_E L Scale 5 0.16 3 SA ref
Transistor_F L Scale 5 0.25 3 SA ref
Transistor_I W Scale 0.25 5 3 SA ref
Transistor_J W Scale 0.4 5 3 SA ref
Transistor_K LW Scale 0.25 0.13 3 SA ref
Transistor_O LW Scale 0.18 0.25 3 SA ref
Transistor_L LW Scale 0.25 0.16 3 SA ref
Transistor_M LW Scale 0.18 0.16 3 SA ref
Transistor_N LW Scale 0.25 0.25 3 SA ref
Transistor_A1 Large 5 5 2 SA 2
Transistor_A2 Large 5 5 1 SA 1
Transistor_B1 Narrow 5 0.18 2 SA 2
Transistor_B2 Narrow 5 0.18 1 SA 1
Transistor_C1 Short 0.13 5 2 SA 2
Transistor_C2 Short 0.13 5 1 SA 1
Transistor_D1 Small 0.13 0.18 2 SA 2
Transistor_D2 Small 0.13 0.18 1 SA 1
Transistor_D1_new Additional 0.13 0.18 2 SA 2
Capacitance DUTs
C_BD_Area BD Area
C_BD_Perim BD Perim
Help

Figure 47 Information Folder during Extraction

DC Initialize

The folder *Initialize* (Figure 48 for BSIM3 and Figure 49 for BSIM4) is intended to set initial conditions for parameter extraction. Since the initial conditions for BSIM3 and BSIM4 models differ, the following section shows both *Initialize* folders, one after the other.

BSIM3

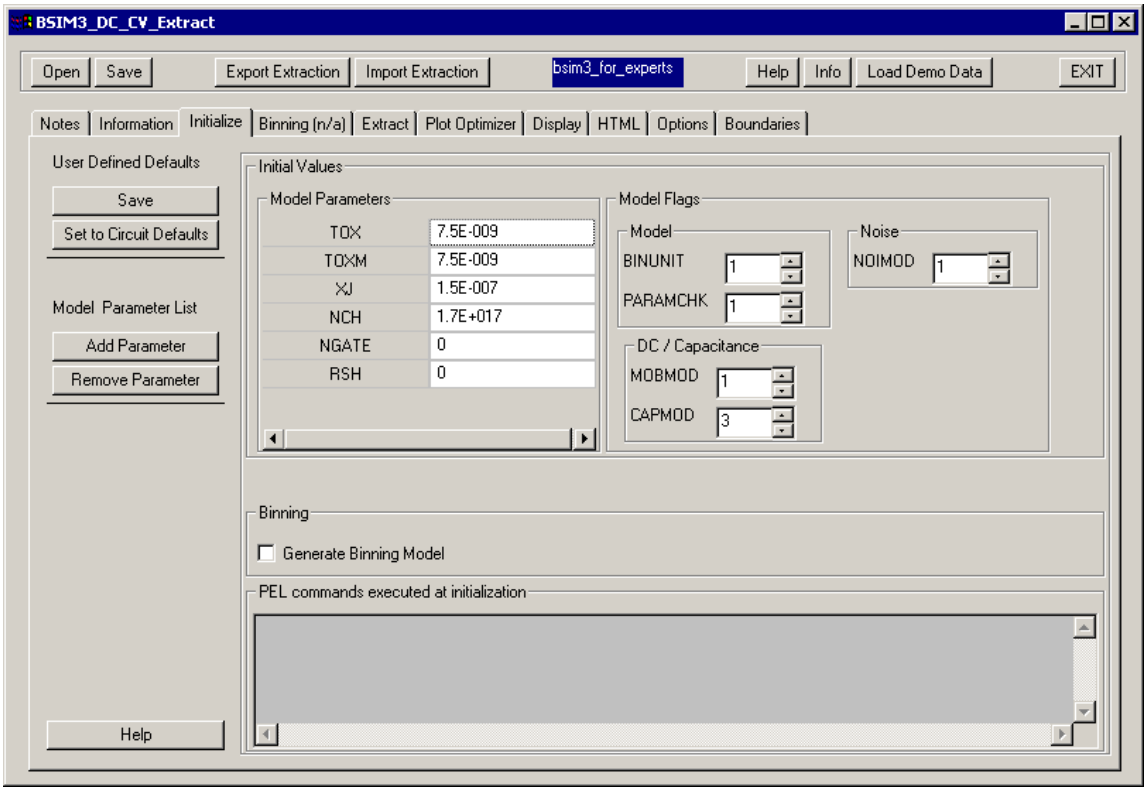


Figure 48 Initialize folder to set initial conditions for the extraction of BSIM3 parameters

The *Initial Values* section of the folder contains fields for *Model Parameters* and *Model Flags*. Into the field *Model Parameters*, enter process related parameters like oxide thickness (TOX) or doping concentrations (channel doping concentration NCH respective gate doping concentration NGATE), and so on. Entering values into the fields and selecting the button *Save* starts a routine to check the values entered. This routine will flag an error message and

change the color of the field whose parameter is given an unrealistic value. The specific field will be marked with red color and remains red until the value is corrected.

The *Model Flags* section is used to set BSIM3 model flags. There are fields for global model flags like BINUNIT or PARAMCHK as well as fields for DC/Capacitance and Noise model flags. The model flags are set to a default value as has been described in the BSIM3 manual from UC Berkeley [1]. See "Model Selection Flags" on page 290 for details or the above mentioned UCB manual [1].

If you would like to use the binning capability, check the *Generate Binning Model* button. With this button checked, the folder *Binning* is activated.

NOTE If you check *Generate Binning Model*, the folder Plot Optimizer will be deactivated. You cannot use the Plot Optimizer together with a binned model.

There is a field provided to enter PEL commands, which are executed at initialization of the extraction process.

To the left, there are buttons under *User Defined Defaults* to *Save* the settings or to *Set to Circuit Defaults*.

Under the *Model Parameter List* section you are able to *Add* or to *Remove a Parameter* to the list shown in the middle of the folder. You will get a list with BSIM3 parameters to select from. This might be helpful if you would like to extract some specific parameters at initialization.

BSIM4

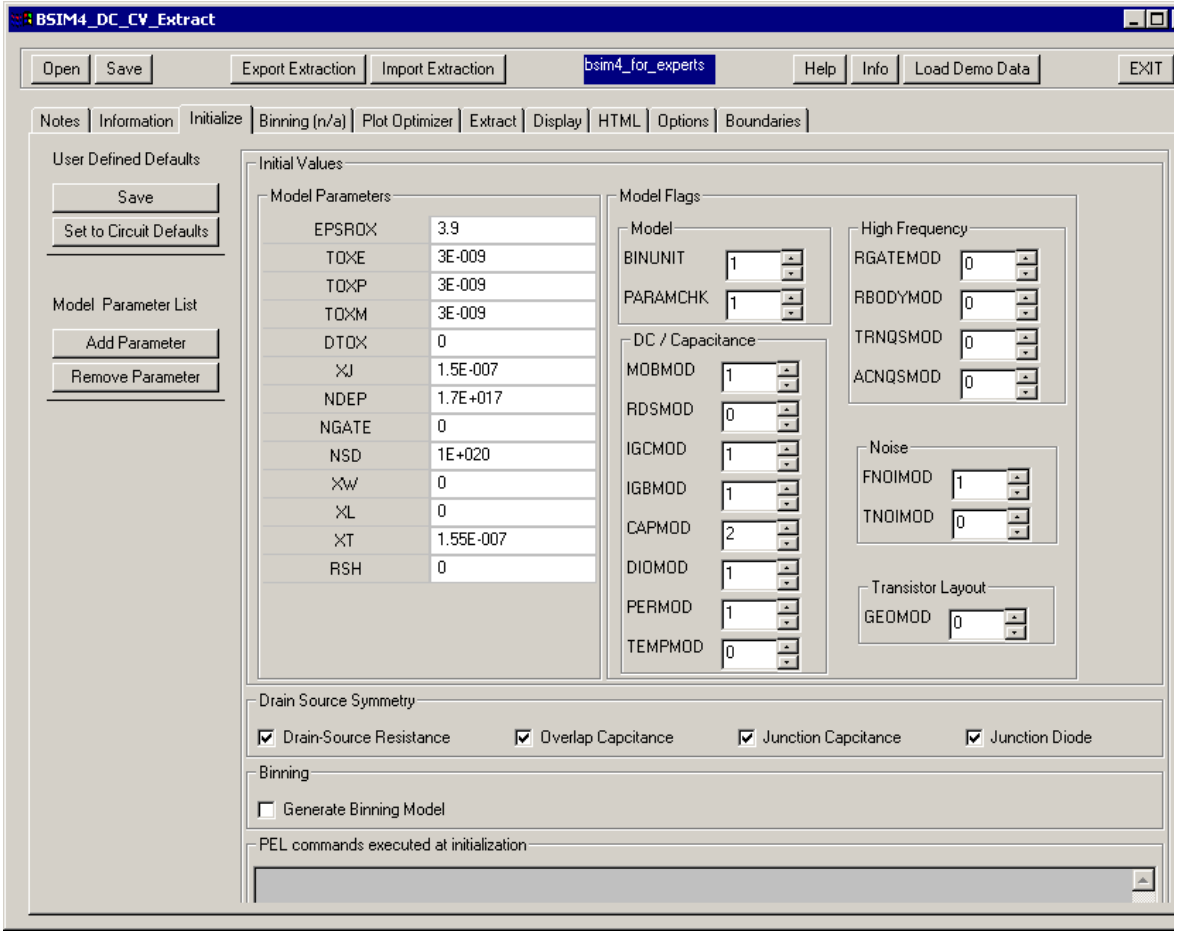


Figure 49 Initialize Folder for the BSIM4 Modeling Package

The *User Defined Defaults* section contains fields for *Save* and *Set to Circuit Defaults*. Inside the *Initial Values*, *Model Parameters* section, enter process related parameters like the relative dielectric constant of the gate oxide, EPSROX. Advanced CMOS process generations are more and more making use of *high-k* gate dielectrics. Therefore, you can

specify the relative dielectric constant of your process by changing EPSROX from 3.9 (default value for SiO_2 gate dielectric). There are other process parameters to be specified on this folder, including electrical, process, or measured gate oxide thickness, TOXE, TOXP, TOXM; junction depth, doping concentrations, and sheet resistances. You will find a description of the model parameters and model flags for the BSIM4 model in "Main Model Parameters" on page 408. See also the manual from UC Berkeley Appendix A: "Complete Parameter List," for more details on model parameters [2]. Entering values into the fields and selecting the *Save* button starts a routine to check the values entered. This routine will flag an error message and change the color of the field whose parameter is given an unrealistic value. For example, if you enter `-3` into the EPSROX field, this field will be marked with red color and remains red until the value is corrected.

You are able to add BSIM4 parameters to the *Initial Values* by clicking *Add Parameter* inside the *Model Parameter List* field. You will be prompted with a list of BSIM4 parameters. Select the parameter(s) you would like to add and click OK. The parameter(s) are added and you are able to enter initial values as desired.

The *Model Flags* section is used to set BSIM4 model flags. The fields only enable settings as defined in the BSIM4 model and are predefined to standard settings.

There is a field defining the symmetry of the drain and source areas. Check the appropriate box(es) if drain and source are processed using the same dose of implantation as well as the same geometry, and therefore the parameters are equal for drain and source areas.

NOTE

Since most MOS processes use symmetric source and drain processing parameters, there is no need to extract the parameters for the bulk-source or bulk-drain diodes separately. Instead, check that the symmetry fields and the respective parameters are set equally.

Only for unsymmetrical processes, which could be modeled in BSIM4, the fields remain unchecked and a separate parameter set will be extracted for bulk-source and bulk-drain diodes.

Binning

You will find some theory on binning inside "Binning of Model Parameters" on page 318. The following figure shows the *Binning* folder, used in binning model parameters. This folder is active only if the flag *Generate Binning Model* is checked, otherwise you will find an n/a sign next to the folder name. This flag is located on the *Initialize* folder under *Generate Binning Model*.

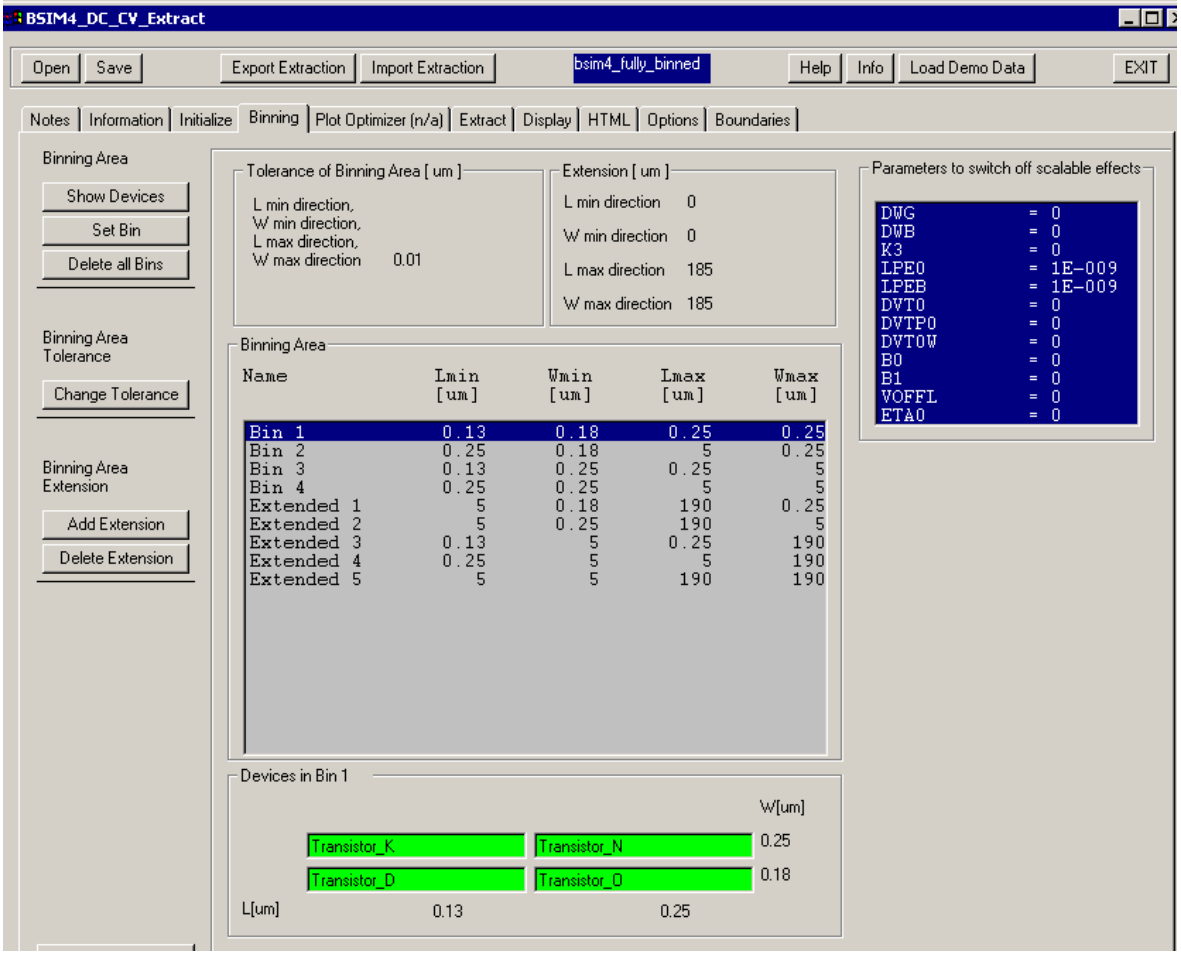


Figure 50 Binning Folder, the green shaded fields name the corner devices of the selected bin (see "Add Extension" on page 149")

Show Devices

By pressing this button, you will see a diagram using logarithmic axes of gate width over length, ranging from 0.1 to 100 microns showing the defined bin boundaries. Inside this diagram you will find markers for existing (measured) devices for this project.

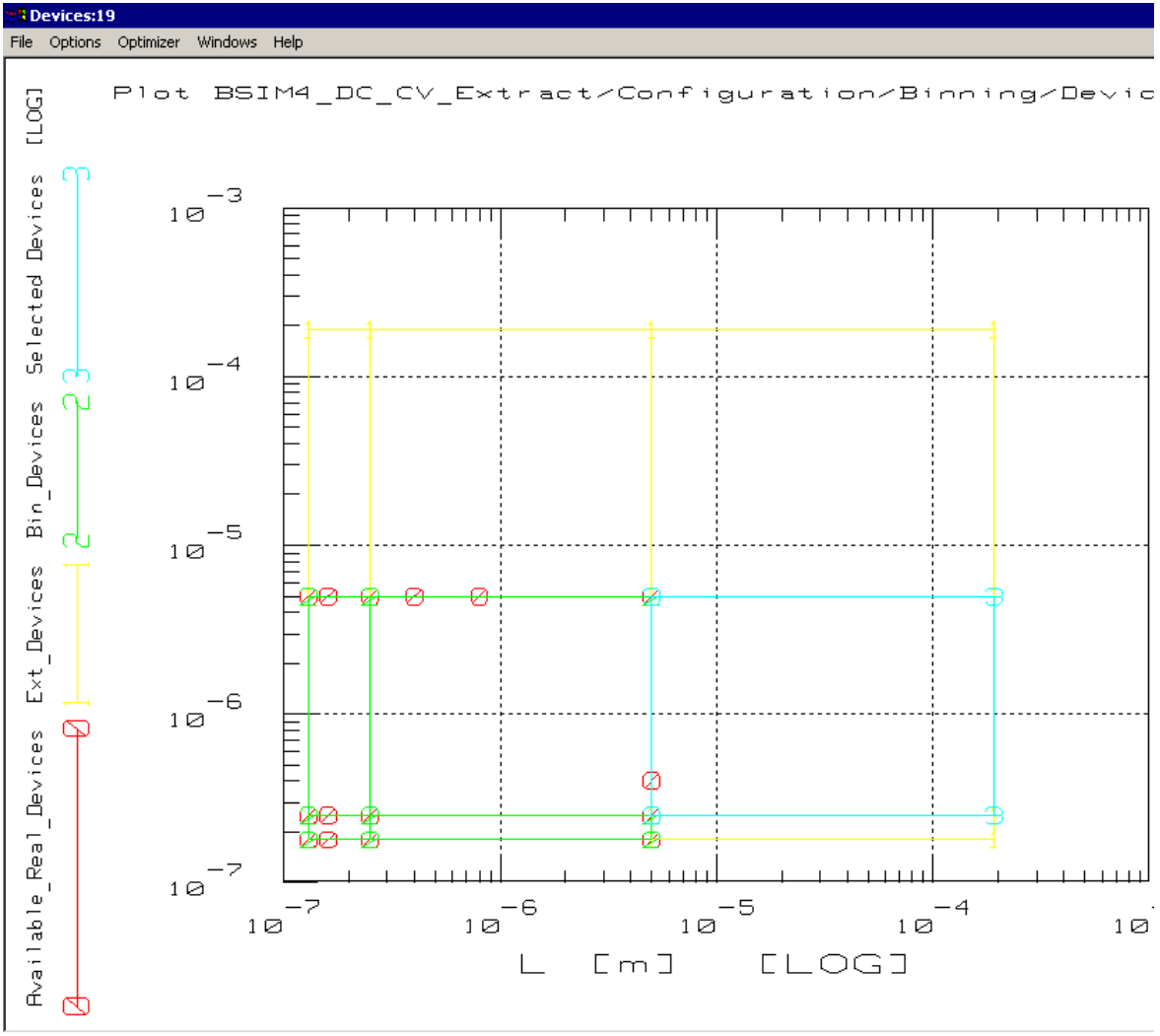
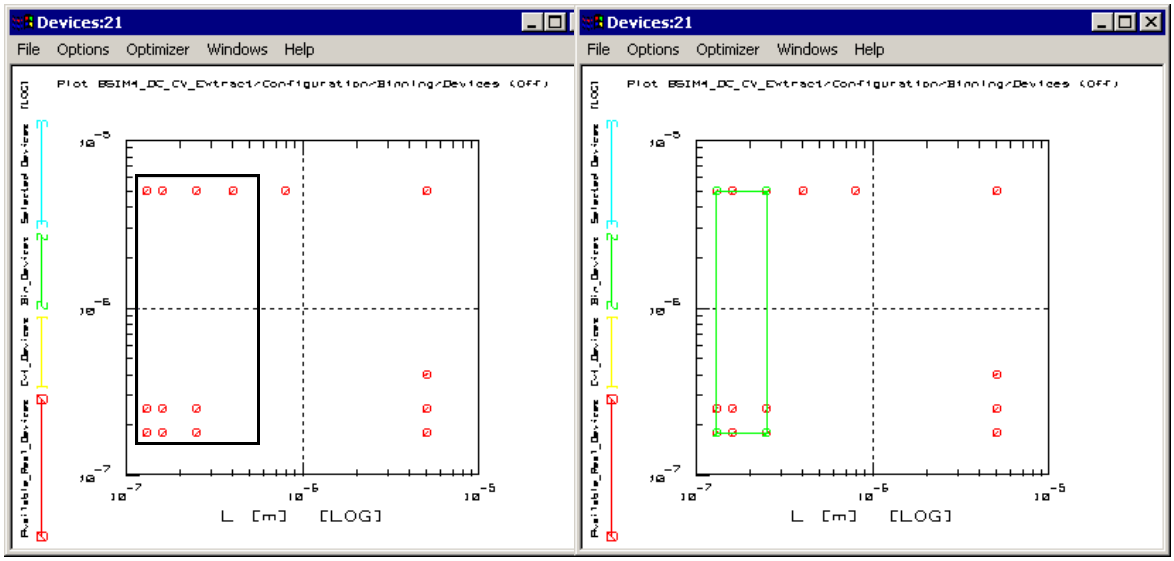


Figure 51 Diagram of measured devices (red), defined bins (green), extensions (yellow) and one selected bin (cyan)

Set Bin

Select bin boundaries by using the displayed diagram, marking two adjacent corners of a rectangle representing the bin and choose *Set Bin*. Be sure to include devices at every corner of your bin, otherwise you will get an error message stating that the selected bin is not rectangular. See the figures below for clarification.



Binning areas:

Left: Not correct, because not every corner of the marked rectangle has a measured device.

Right: Correctly defined binning area.

The selected binning areas are automatically entered into the form using a bin number and the geometries of the four corners for this bin.

Delete all Bins

Using this button, all bins are deleted from the graphic as well as the form.

Change Tolerance

The purpose of this button is to change a predefined tolerance for the binning areas. This tolerance is needed because of the definition of Binning Boundaries.

There are boundaries for each binning area: LMIN, LMAX, WMIN, WMAX. Those boundaries will be analyzed using Leff and Weff. If $LMIN(bin1) \le L < LMAX(bin1)$, a subcircuit will be used. This means, if a device with L=10µm is used and this is LMAX, then it is not possible to simulate this device using the binned model. This is due to the above mentioned region for the parameters: $LMIN(bin1) \le L < LMAX(bin1)$. So if L = LMAX, the device does not fit into the binning boundaries (which require a value smaller than LMAX) and cannot be used for simulations.

Therefore, tolerances are implemented to correct for this error. You are able to change the predefined tolerance (0.01µm) to a value which suits your needs. The results within the measured and extracted areas will not be altered. But it is now possible to simulate devices having a gate length or width a little delta L or delta W outside the defined binning areas.

Add Extension

By choosing this button, you will get a form to enter Extension Delta Values like the one shown below.

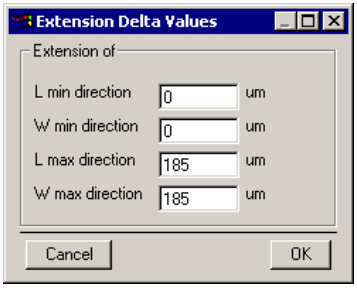


Figure 52 Extension values form

If the extension is not activated, certain simulators would not be able to simulate devices with L=Lmax or W=Wmax of certain bins. The extension delta values define the extensions from the measured devices. This means, you must set extension delta values Lmin and Wmin within the range of the minimal measured device, otherwise you will get an error message. In other words, if your minimal measured device uses a gate length of 0.15µm, the extension in Lmin direction must be set between 0 and 0.149µm. There is no limit for the extension in the Lmax and Wmax direction.

If you select one of the defined bins, the fields under Devices in Bin<No> will become green shaded and will show the name of the corner devices of this bin and the corner geometries, see Figure 50. At the same time, the diagram will show the selected bin boundaries in light blue color.

Delete Extension

You can delete the entered extensions by choosing the *Delete Extension* button.

The field *Parameters to switch off scalable effects* is used to set which parameters use the scalable possibilities as defined inside the BSIM4 model and which parameters are prevented from scalable modeling in BSIM4. All deselected parameters are using the extracted values, whereas all selected parameters (marked with blue background) are using default values for binning purposes. The parameters DWG and DWB are always off, therefore they cannot be de-selected.

DC Extract

The next folder, **Extract**, defines the Extraction Flow for the devices. There is a standard extraction flow implemented, but you can change this flow if you find another one suiting your needs better than the default one.

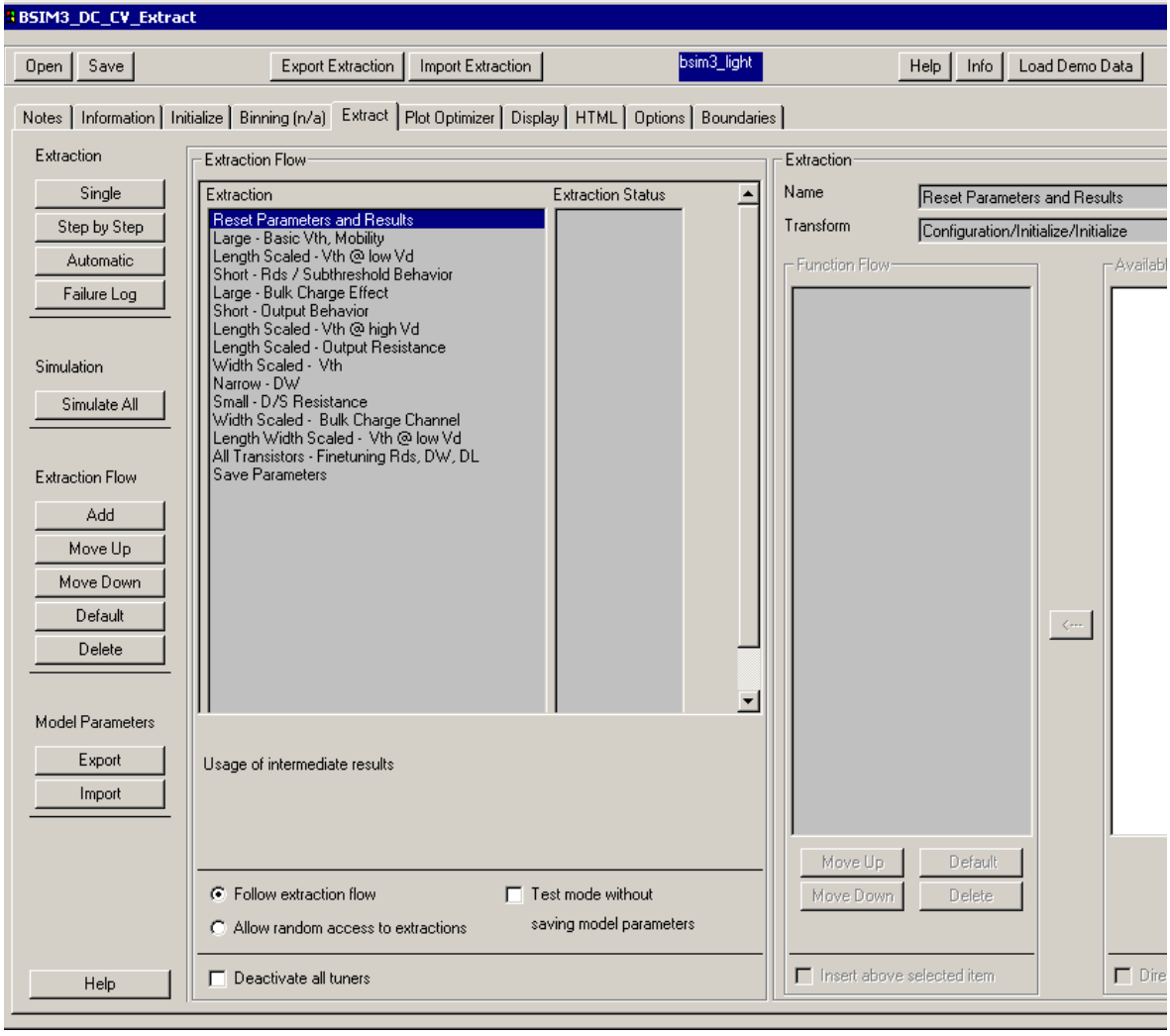
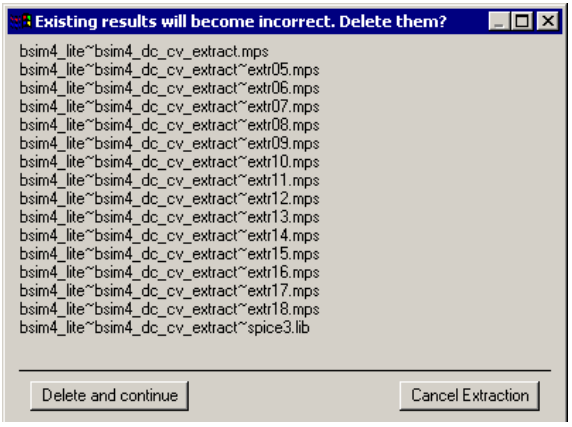


Figure 53 Extract form

The folder shows fields named *Extraction Flow*, *Extraction Status*, *Function Flow*, and *Available Functions*. The *Extraction Flow* field shows the name for the selected extraction step. Under *Function Flow*, the functions used for the selected extraction flow are listed. The *Available*

Functions field shows a list of functions to be used for a selected extraction flow. Mark the desired function and click the arrow in between the *Function Flow* and *Available Functions* fields to add the function to the flow list. The letters in front of the function name explain the extraction method used for this function: E represents *extraction*, O stands for *optimization*, and T for *tuning*.

To the left, there are buttons to *Add* or *Delete* Extractions from the flow. You will be prompted with a list of available extractions. Select one of the extraction steps and press *Add* on that form. If you have already extracted parameters, the Extraction Status field shows intermediate steps. It will be shown, which results are being used for the selected extraction step. Now, if you *Add* or *Delete*, change the arrangement of the steps or reset to *Defaults*, the following extractions will become invalid because they are based on the results of the extraction immediately before them. Therefore, you will be warned before changes are made!



Arranging the flow is possible by using the *Move Up* or *Move Down* buttons to the left of the folder.

To extract parameters from one flow only: Select the desired flow under the Extraction Flow section in the left half of this folder and choose the *Single* button. Again, you will be warned before the selected extraction will be performed.

To go through the extraction process one step at a time, highlight the step and choose *Step by Step*. A dialog box may appear, prompting you for input.

- To automatically extract all parameters using the extraction flows listed under the Extraction Flow section, choose *Automatic*. The programmed Extraction Flow will be extracting all parameters defined in the active extraction flow list. The programmed extraction flow has to begin with the *Reset Parameters* step, otherwise you will get an error message.
- In case you would start an extraction of some parameters after you already have extracted some other parameters, you are not able to start from the beginning without resetting all parameters, including the ones already extracted. To re-extract or to optimize one parameter after some other parameters are already extracted, simply add the desired step in the extraction flow list on a place further down the list. In that case, the extraction process uses the parameters already extracted during an earlier step in the extraction process and you overcome the reset parameter step.
- All warnings and errors during the extraction process are written to the failure log, which is opened using the *Failure Log* button.

NOTE The contents of the Failure Log window doesn't contain all the warnings written to the IC-CAP status window. Only the warnings and errors regarding parameter extraction are re-directed to the Failure Log.

- The button *Simulate All* starts simulating the circuit as defined in IC-CAP.

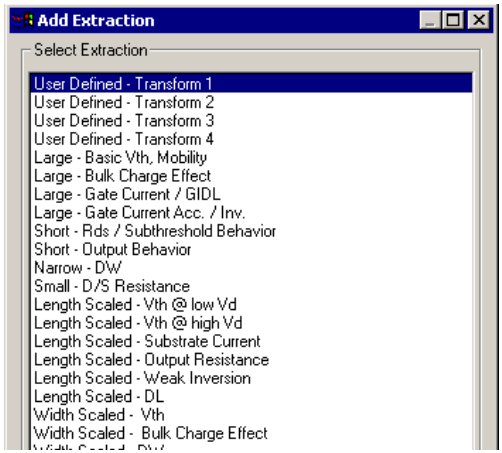

Add Extraction
Select Extraction
User Defined - Transform 1
User Defined - Transform 2
User Defined - Transform 3
User Defined - Transform 4
Large - Basic Vth, Mobility
Large - Bulk Charge Effect
Large - Gate Current / GIDL
Large - Gate Current Acc. / Inv.
Short - Rds / Subthreshold Behavior
Short - Output Behavior
Narrow - DW
Small - D/S Resistance
Length Scaled - Vth @ low Vd
Length Scaled - Vth @ high Vd
Length Scaled - Substrate Current
Length Scaled - Output Resistance
Length Scaled - Weak Inversion
Length Scaled - DL
Width Scaled - Vth
Width Scaled - Bulk Charge Effect

Figure 54 Adding an Extraction Step

You can add several steps of the same extraction after each other. The extraction method selected in one step could be another one in a further extraction step. In other words, you are able to set the tuner option in one extraction step and the optimizer option in another step, probably when other parameters of influence have been extracted in between the steps.

- Change the function flow inside the *Function Flow* field using the *Move Up* or *Move Down* buttons below that section to move a selected extraction routine one step up or down.
- The *Default* button restores the order of parameter extractions inside the *Function Flow* list as it was in the beginning of a project.
- To delete an extraction from the *Function Flow*, choose *Delete*.

NOTE

You cannot delete the first (Reset Parameters) or the last (Save Parameters) extraction step inside an extraction flow.

- Below the Extraction Flow field, check boxes enable you to *Follow Extraction Flow* as programmed or to use the *Test Mode* without saving the extracted parameters. This mode is intended for you to test influences of some parameters on others without overwriting already extracted parameters from the selected step. You are able to use every *.mps file with every extraction step without saving and resetting parameters.
- To access extractions randomly, check *Allow random access to extractions*. With already extracted parameters, you will be prompted to delete the extracted parameters and start with new ones.

You will notice the changing of colors on the *Extraction Folder*. This is a visual warning that intermediate results are not stored in this mode!

NOTE

This feature makes it possible to test influences of any other *.mps file to compare extraction results.

Follow extraction flow means: An extraction step uses the results of the step above, as is stated under *Usage of intermediate results* in the middle of this folder!

Test mode starts with the last step taken in the flow. No results are overwritten or saved!

Allow random access to extractions: Existing .mps files are being deleted! The extracted parameters are continuously updated inside ICCAP, but not saved. You are able to perform extractions in any desired sequence. The *Extraction Status* field shows the sequence of extractions! In this mode, you are able to use the Test mode as well. On leaving the Test mode, the previous state of extractions will be reloaded.

- Using the *Direct Execution Mode* button below the *Available Functions* field (possible only in *Test* Mode or in *Random Access* Mode) changes the color of the field and the header will now read: *Executable Functions*. If you click on one of the executable functions now, this function will be executed as it is defined. This means, if the function is an extraction step, the selected parameter for this function is extracted. If it is a tuning step, the IC-CAP tuner opens with the selected parameter sliders, and if it is an optimization step, the optimizer for this function will be executed.
- To export the extracted parameters during the defined extraction flow for the purpose of saving intermediate results, use *Export* under the *Model Parameters* section. You must specify the path and the name for the parameter file to be exported. Exported files will be packed into a .tar file using the project name as a file name and appending *Export* and a number to the extracted .tar file. (*~project_name~Export_1*, for example)
- Parameters can be saved by selecting *Save Parameters* from the extraction flow and *Single* from the menu buttons on the left, only if all extractions are done.

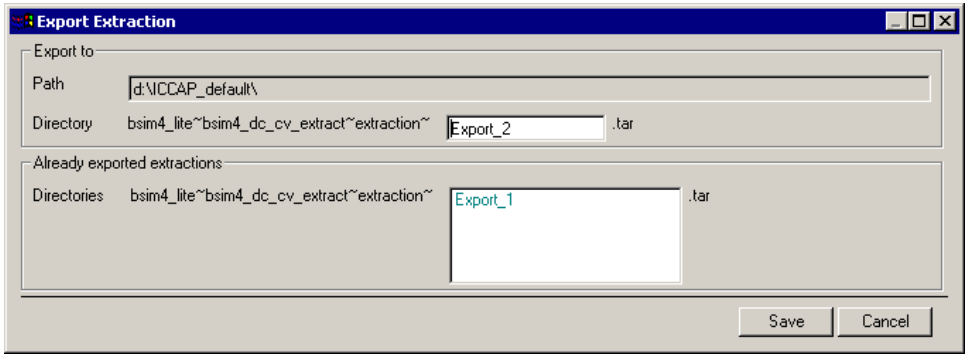


Figure 55 Export Model Parameters

- A button is provided to *Import* parameters, for example, from an earlier project. You will receive a warning message stating that importing parameters will overwrite the actual model parameters in IC-CAP.

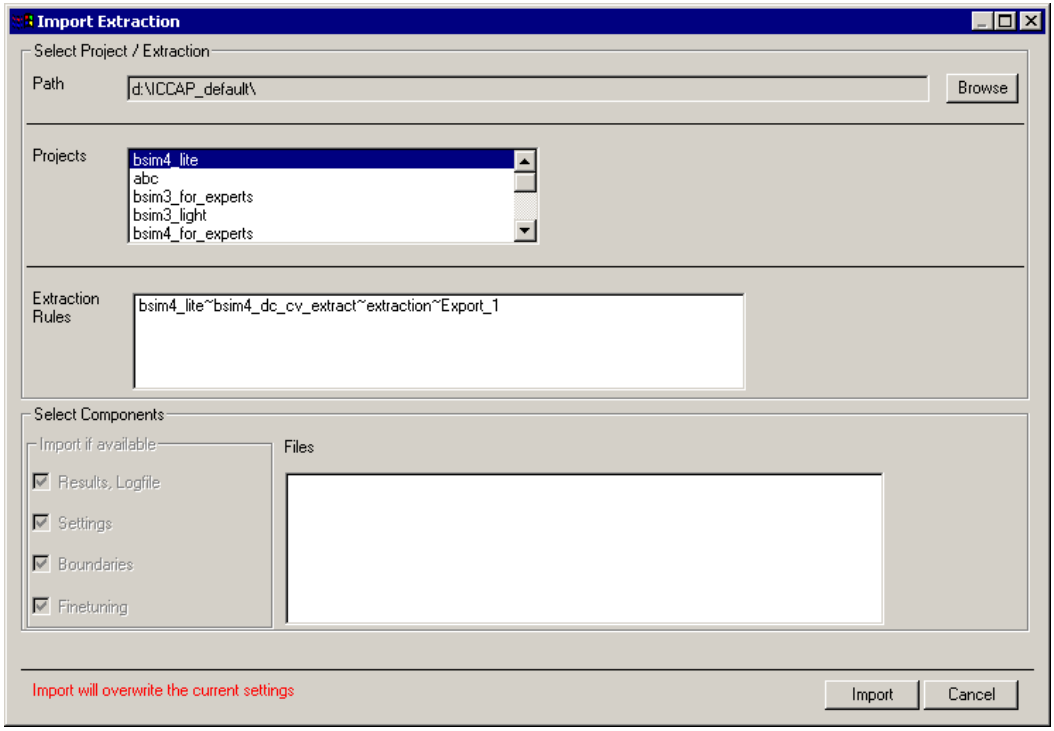


Figure 56 Import Extractions

Select either a Project or, if existing, an exported .tar file for a specific project. In this case, you must select the project as well as the exported extractions. Then you will be able to select whether you would like to use all or specific parts of the saved extraction project components by de-activating the components not to be used (results, settings, boundaries and finetuning). If there are stored files, they will be shown under the *Files* section of this window.

NOTE

This feature makes it possible to import any other *.mps file to compare results. But be careful, importing other parameter files will overwrite the actual parameters. Don't use this option during an extraction session with partly extracted parameters unless you've saved the work in progress!

Plot Optimizer

This folder is active only *without* the *Generate Binning Model* check box on the *Initialize* folder selected.

This folder enables you to set up multi-transistor targets for a global optimization of parameters. The BSIM3/4_DC_CV_Extract module allows the definition of different optimizer tasks. An optimizer task is composed of different devices and different possible setups per device. For example, an optimizer task to fine tune the drain saturation current versus gate length would need different short channel devices and the idvd and idvg_vdmax setup per device. This configuration can be done inside the *BSIM3/4_DC_CV_Extract* GUI, folder *Plot Optimizer*. The resulting *BSIM3/4_DC_Finetune* model will be saved within the actual project directory. A template model file is in *$ICCAP_ROOT/examples/model_files/mosfet/BSIM3(4)/utilities*, which is called BSIM3(4)_DC_CV_Finetune.mdl. This template will be loaded, if you click *Open Model* in the *Load Finetuning Model* section to the left of the *Plot Optimizer* folder.

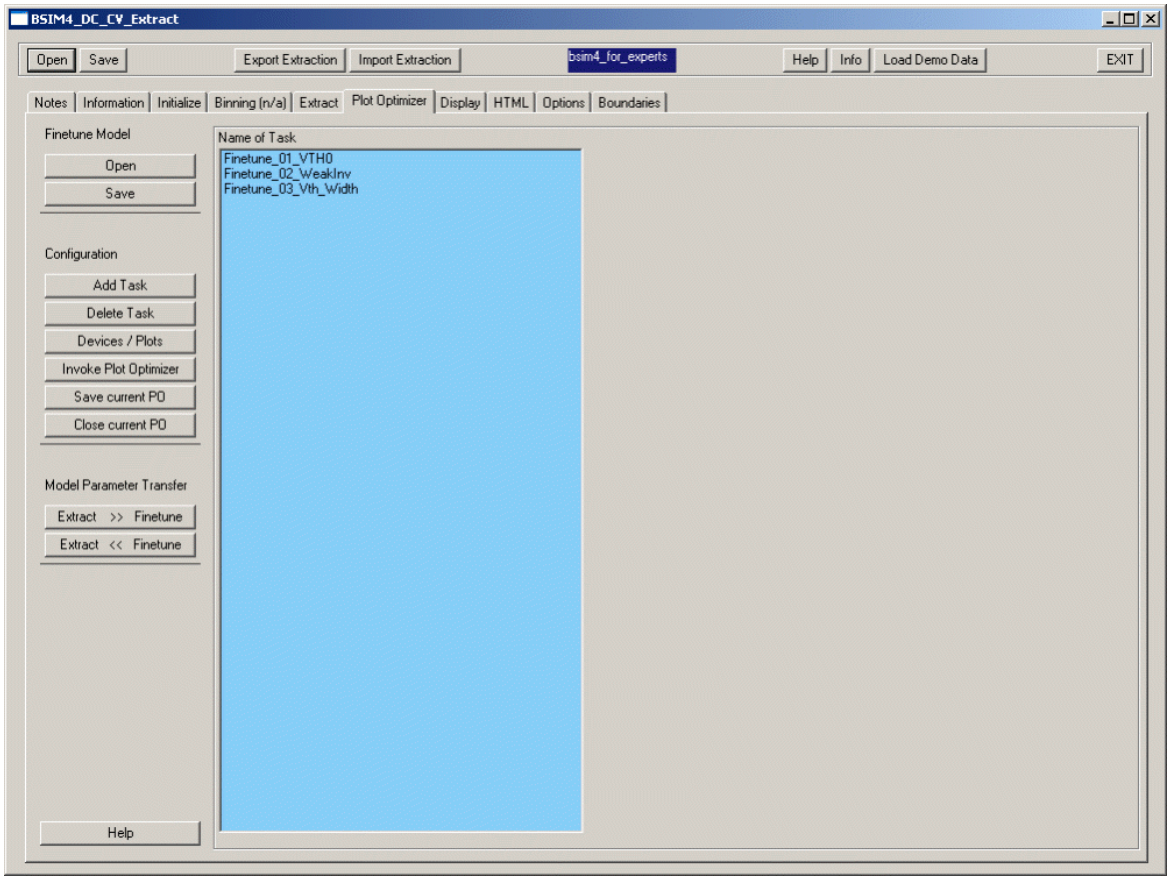


Figure 57 Plot Optimizer folder

Each setup in this DUT is a combination of a device and a certain simulation setup. The setup name will be <device>_<setup> (e.g., Transistor_A_idvg). The appropriate device dimensions for a certain device are referenced by the input "*index*" in each setup which sets a variable "INDEX" to act as a pointer to the model variables DUT_L, DUT_W, ... Those variables contain the gate lengths and widths of all devices in this actual project.

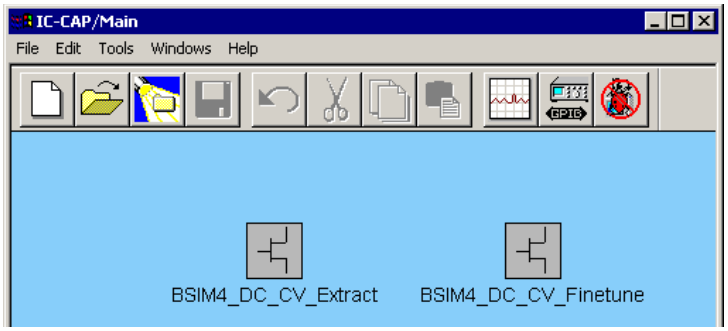
You can select from a set of predefined setups (idvg, idvg_vdmax, idvd, dvd_vbmax) or you can easily define your own setups. For details, see item “User defined setups” on page 163.

Setting up an optimizer task

To set up an optimizer task, please perform the following steps:

Load the Fine-tuning model by clicking *Open*.

Clicking *Open Model* opens a new model file inside the IC-CAP/Main window. This model file is called *BSIM4_DC_CV_Fine-tune*.



Configuration of fine-tuning tasks using the BSIM3/4_DC_Extract model

Under the *Configuration* section of the *Plot Optimizer* folder in BSIM3/4_DC_Extract, click *Add Task*. You will be prompted to enter a name. A *FinetuneConfiguration* window will open, which is shown in Figure 58. This figure shows the configuration process for a certain fine-tuning task. You have to select which Setup/Plot to use for which device. After configuring the fine-tuning model, click *OK* to close the form.

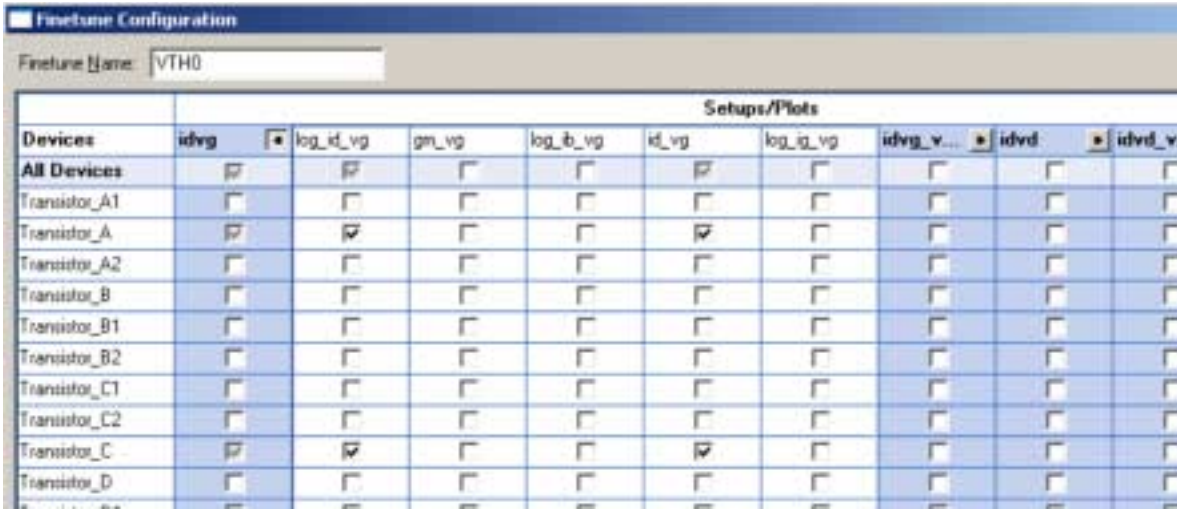
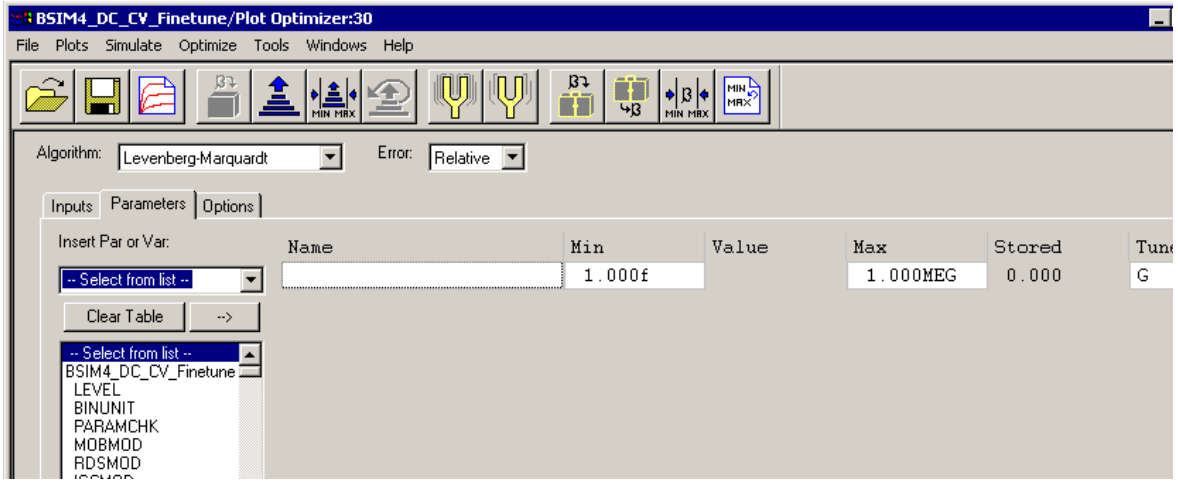


Figure 58 Configuring a fine tuning task

At the same time, the Plot Optimizer window opens, as is shown in the following figure.



Select the desired fine-tuning configuration by activating the desired Setup or Plot for a certain device. By clicking the arrow in the top row beneath the Setup/Plot name (idvg, for

example), you are able to select which of the predefined plots you would like to be included into the fine-tuning configuration. Close the form by clicking OK. Now, the selected plots are open-end and you have to select the regions used for optimizing. See “Using the Plot Optimizer” in the *IC-CAP User’s Guide* for help using the plot optimizer.

Once the plot optimizer tasks are set up and saved, you can perform an initial manual optimization by moving the actual parameter set from the *BSIM3/4_DC_CV_Extract* model to the *BSIM3/4_DC_Finetune* model or vice versa using the *Extract* >> *Finetune* or *Finetune* >> *Extract* buttons. The fine-tuning tasks can now be added like any other extraction step using *Extraction Flow* > *Add* on the *Extract* folder. The *Add Extraction* form now contains the configured fine-tuning task for selection. After selecting the task, the *Extract* folder in the *BSIM3/4_DC_CV_Extract* model contains the specified plot optimizer tasks. You can invoke them from this folder by clicking *Extraction* > *Single*, if you want to perform a separate optimizer task. On the other hand, you can invoke a plot optimizer task during the complete extraction process by clicking *Step by Step*. It will then be inserted into the programmed extraction sequences.

If one element, for example, *Finetune1* is activated in the extraction flow, all existing plot optimizer transforms in the setup */BSIM4_DC_Finetune/Finetune1_<name>/Config_Setup* will be invoked. See Figure 59 for an example of the *Extract* folder after configuration of a fine-tuning task.

Customizing the plot optimizer

Adding error curves to the diagrams

Error curves as well as the values of maximum error and RMS error between measured and simulated curves can be added to standard plots. Turn on the DUT variable *SHOW_ERROR* to invoke this feature. The function *BSIM4_error()*, which calculates the error values can handle boundaries for error calculation. This may be useful to exclude, for example, measurement noise or to focus the

error calculation to a certain area of operation for a device. The boundaries for error calculation are defined inside the DUT variables of the appropriate DUT. An example for such a set of variables are:

- *X_LOW_IDVG_ID*: minimum usable value of ID in all idvg setups for error calculation
- Y_HIGH_IDVG_IG: maximum usable value of IG in all idvg setups for error calculation

Please see the example transforms for a detailed description of using this feature.

User defined setups

Besides the standard setups in the MASTER DUT, you can add additional setups taking into account the following items:

- The new setup must be placed inside the MASTER DUT, it can be used during configuration.
- The setup must be a subset of the measured data.
- Two setup variables must be set:
  - *NameSetup*: Set this variable to the name of the actual setup. For an example, see *userdefined_example*.
  - *DUT_Type*: This variable tells IC-CAP which type of .mdm file should be used to add.

    The possible choices are "~*dc_idvg*~" or "~*dc_idvd*~".

Configuration of the Plot Optimizer Task

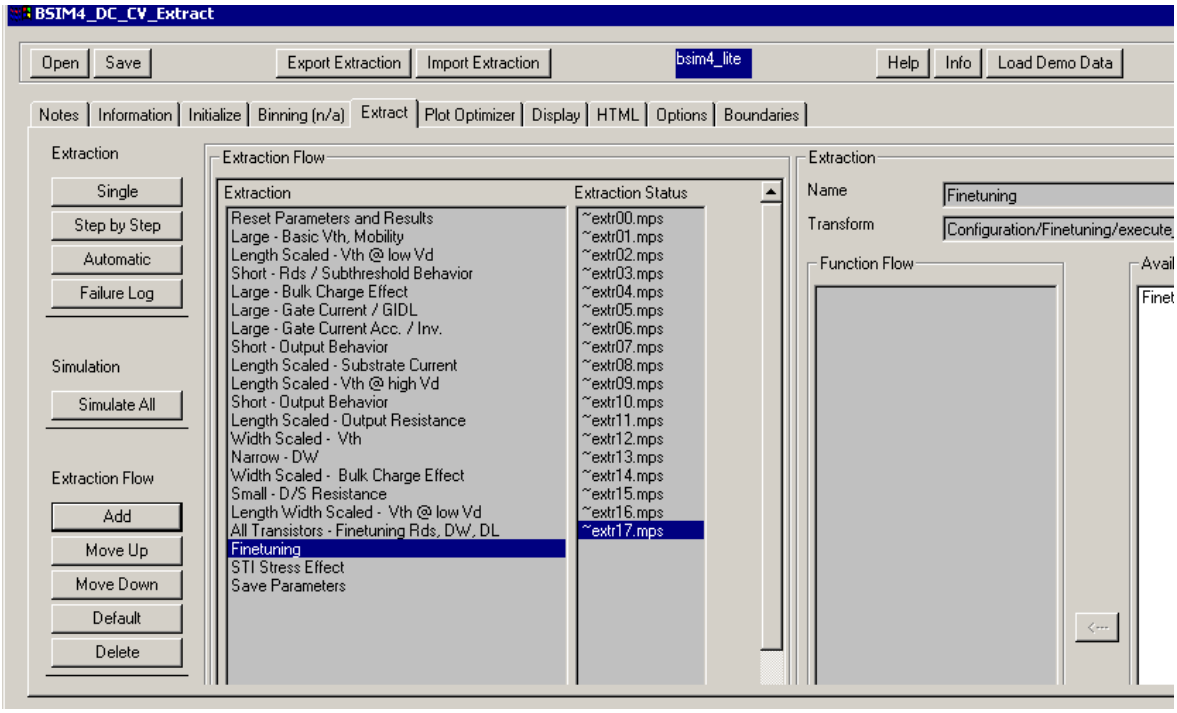


Figure 59 Extract folder after setting up a Plot Optimizer task

DC Display

The next folder to the right, *Display*, is used to display diagrams of measured and simulated setups together to view the results of parameter extraction. It has sections for DC Transistor, Capacitance, and DC Diode behavior. Within those sections, there are fields representing diagrams and their relevant parameters. Click on one of the predefined lines to simulate the selected setup using the voltages defined in *Measurement Conditions* and to display in diagrams.

NOTE The environmental settings regarding size and background color are effective when displaying diagrams using the Display folder.

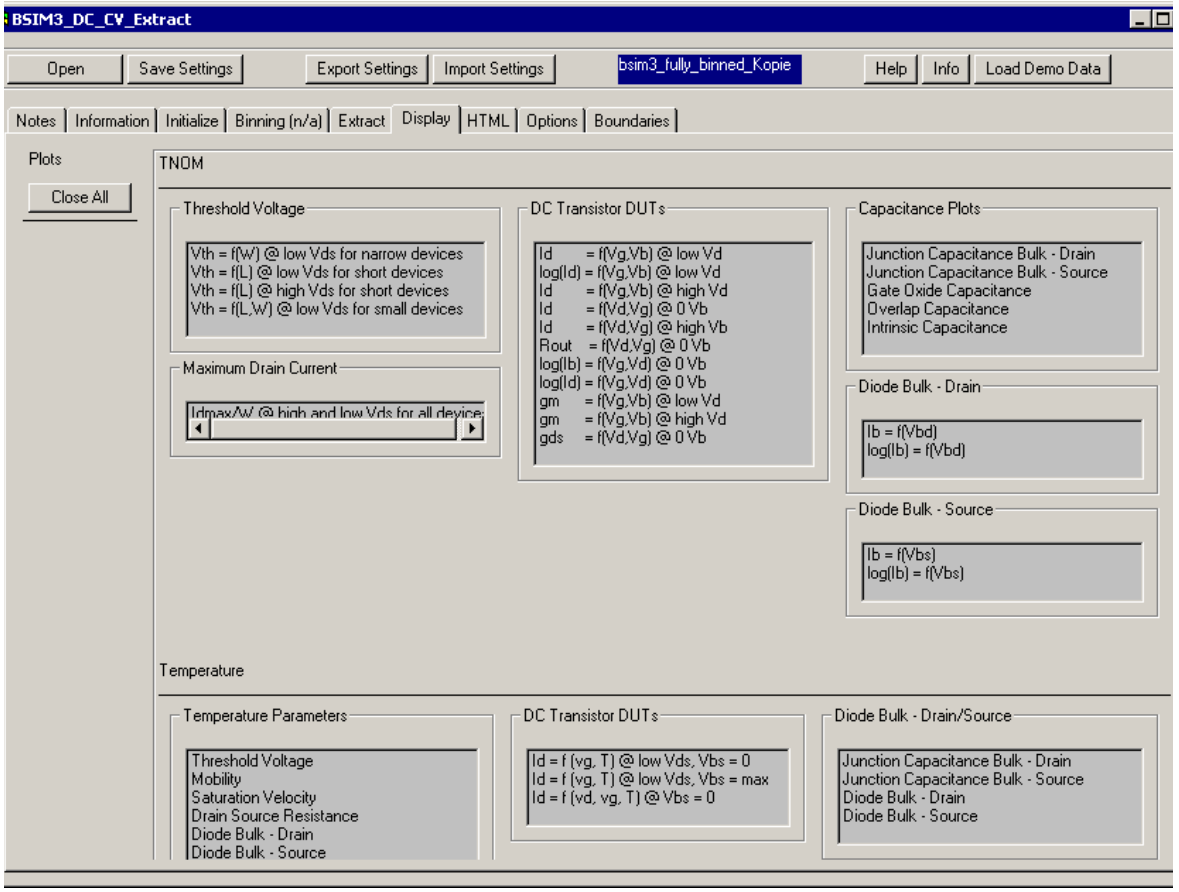


Figure 60 Display simulation results using the parameters extracted

DC HTML

The folder HTML is used to generate a report file in HTML format. You can define a headline and comments for the report, specify the path to save the report, as well as the

command to start the browser. You can also define the size of plots and the diagram background as it appears in the HTML report.

NOTE Generate HTML uses the project mps file (project_name~bsim*_dc_cv_extract.mps), not the loaded or imported one.

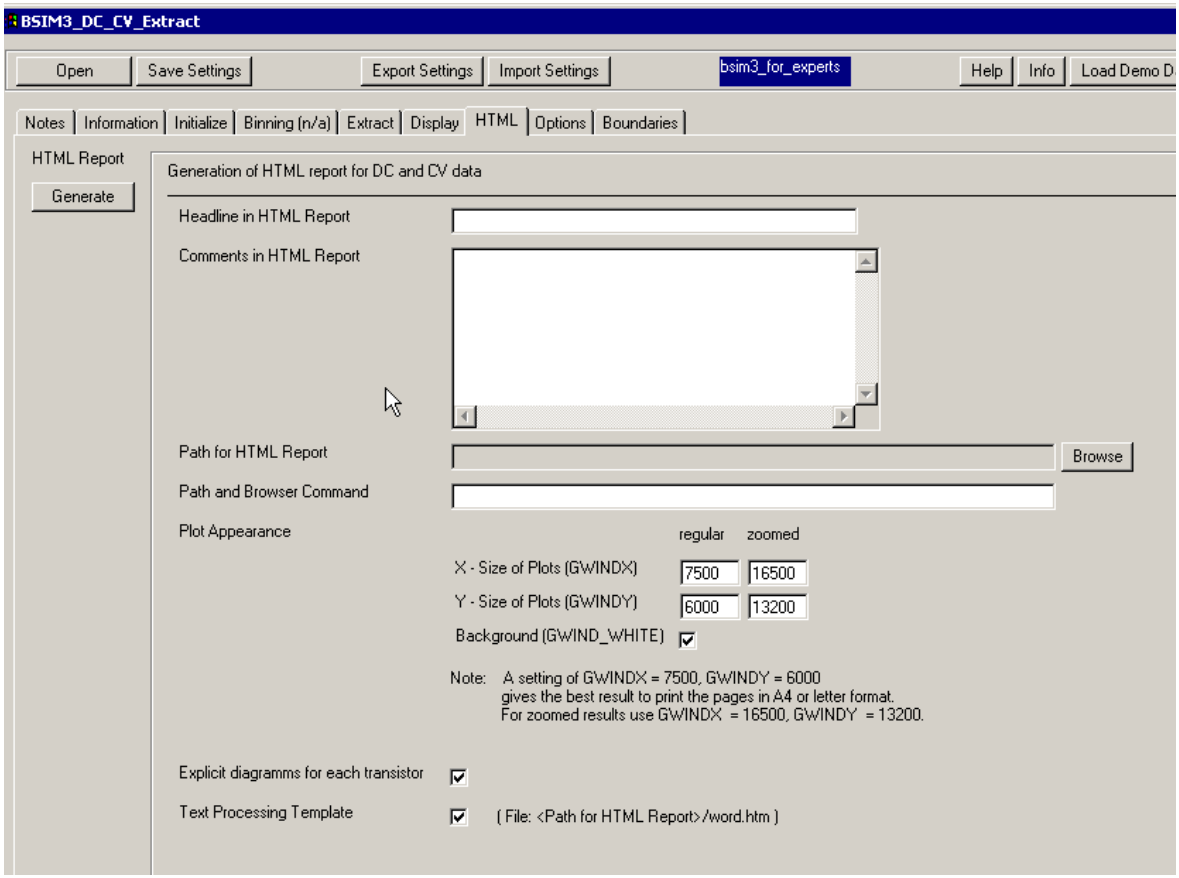
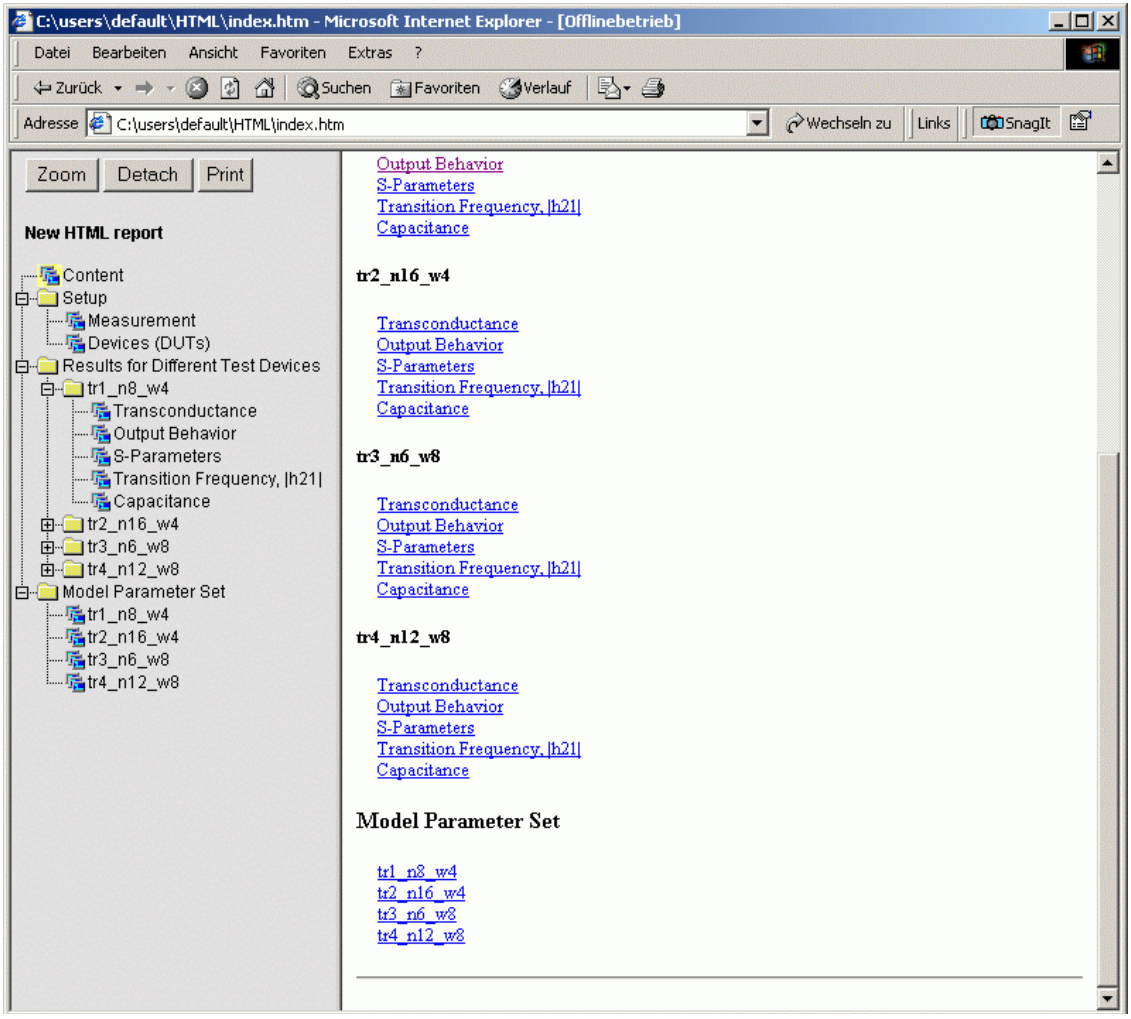
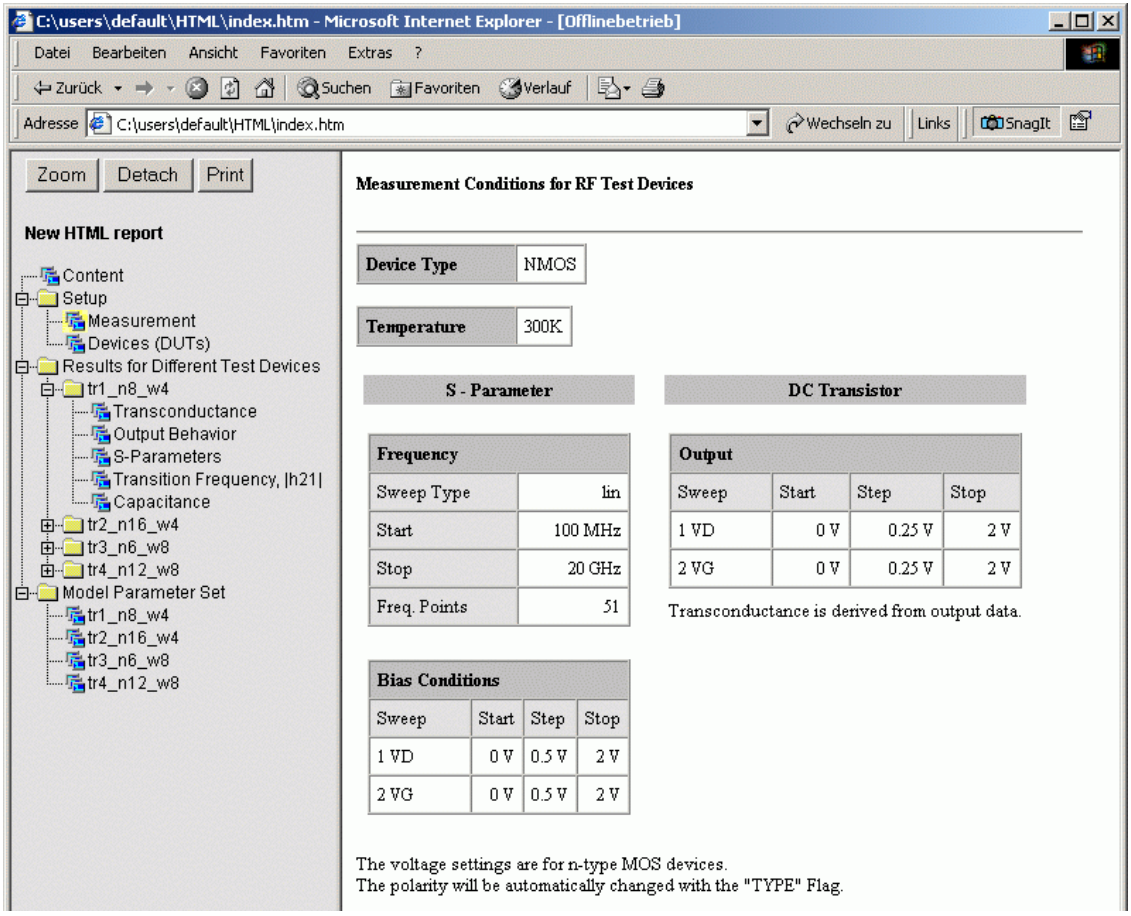


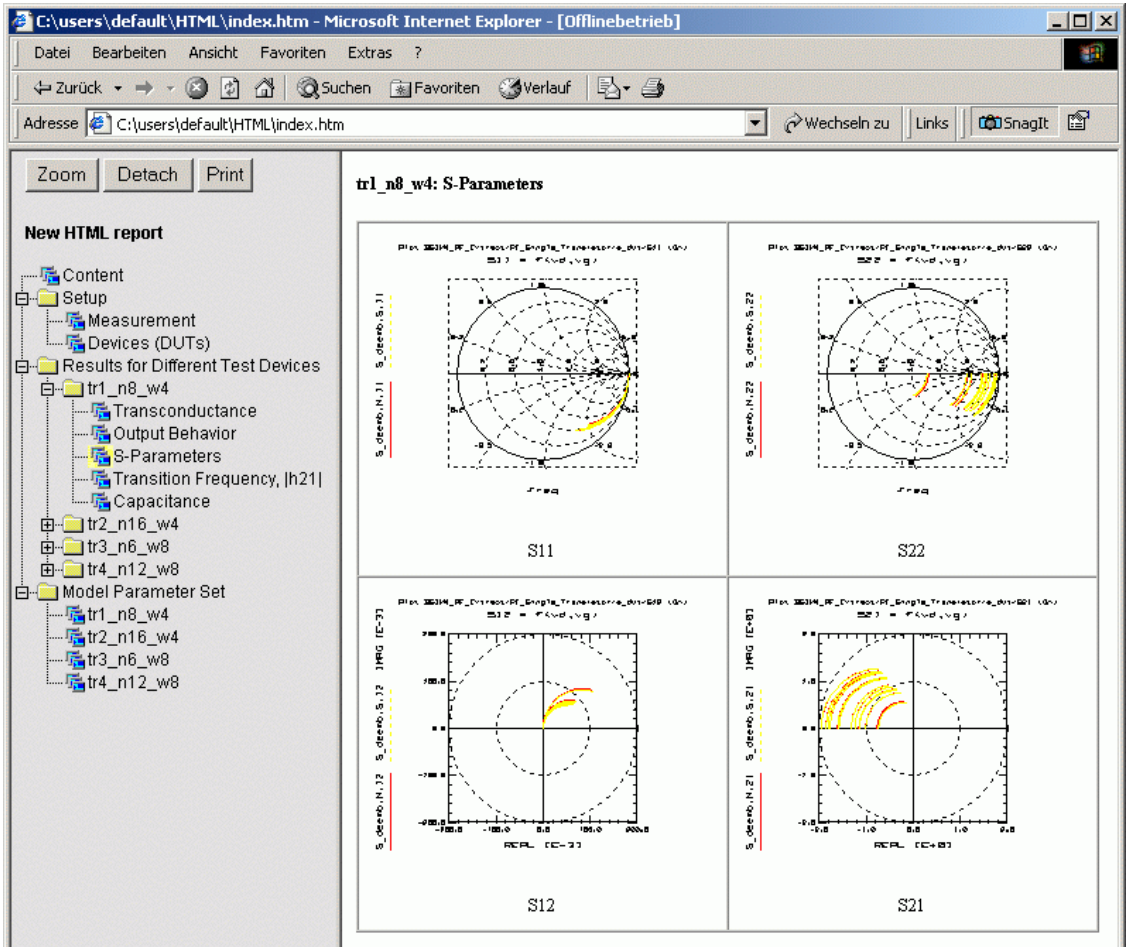
Figure 61 Folder to generate reports in HTML format

If you use a path where an HTML project report already exists, you will get a warning. If the path doesn't exist, you will be a prompted to accept creation of the specified directory.

The following figures show part of a generated HTML report. This report could be published over the intranet for use inside your company or over the web for customer use.



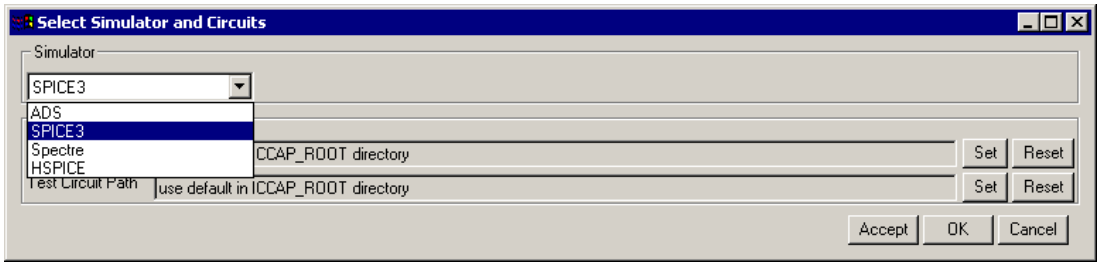

C:\users\default\HTML\index.htm - Microsoft Internet Explorer - [Offlinebetrieb]
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Verlauf
Adresse C:\users\default\HTML\index.htm
Wechseln zu Links SnagIt
Zoom Detach Print
New HTML report
Content
Setup
Measurement
Devices (DUTs)
Results for Different Test Devices
tr1_n8_w4
Transconductance
Output Behavior
S-Parameters
Transition Frequency, |h21|
Capacitance
tr2_n16_w4
tr3_n6_w8
tr4_n12_w8
Model Parameter Set
tr1_n8_w4
tr2_n16_w4
tr3_n6_w8
tr4_n12_w8
Measurement Conditions for RF Test Devices
Device Type NMOS
Temperature 300K
S - Parameter
DC Transistor
Frequency
Sweep Type lin
Start 100 MHz
Stop 20 GHz
Freq. Points 51
Output
Sweep Start Step Stop
1 VD 0 V 0.25 V 2 V
2 VG 0 V 0.25 V 2 V
Transconductance is derived from output data.
Bias Conditions
Sweep Start Step Stop
1 VD 0 V 0.5 V 2 V
2 VG 0 V 0.5 V 2 V
The voltage settings are for n-type MOS devices.
The polarity will be automatically changed with the "TYPE" Flag.



DC Options

The folder *Options* lets you define some environmental conditions used in extraction.

You can set the Simulator used by selecting the *Change* button on the left side of the window. You will see a window like the following one:



You can select which simulator to use from a pull-down list of ADS, SPICE3, Spectre, or HSPICE. You can also select the path to the appropriate circuit files respective the test circuit files. Usually, you will find those files in: ICCAP_ROOT/examples/model_files/mosfet/bsim3(or bsim4) /circuits/SIMULATOR/cir(tci).

NOTE If you would like to modify the standard circuit/test circuit files, be sure to copy the directory ICCAP_ROOT/examples/model_files/bsim3(or bsim4)/circuits/<SIMULATOR> and change the files inside the copied directory, not the original ones.

There are predefined values for the variables. You can change those variables or accept the values. Since some variables in BSIM4 are not used in BSIM3, the folders differ from each other. See the following two figures.

Using those variables you can define a minimum usable current for extraction. The purpose of these variables is to cut out noisy current measurements by defining the lower limit of currents used for extraction of different parameters.

You can specify a printer command as well as the size of plot windows and the background color of those windows.

BSIM3 — Options

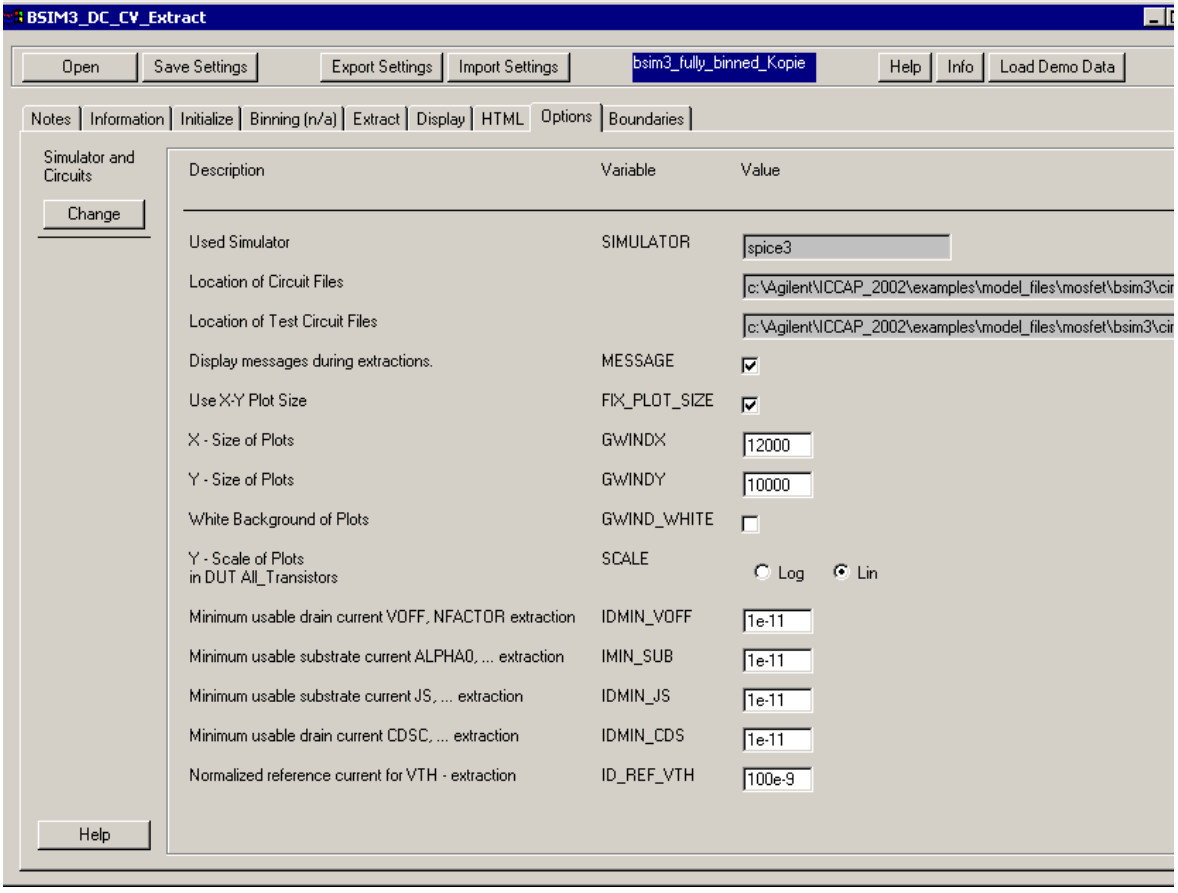


Figure 62 Options used in parameter extraction for the BSIM3 model

BSIM4 - Options

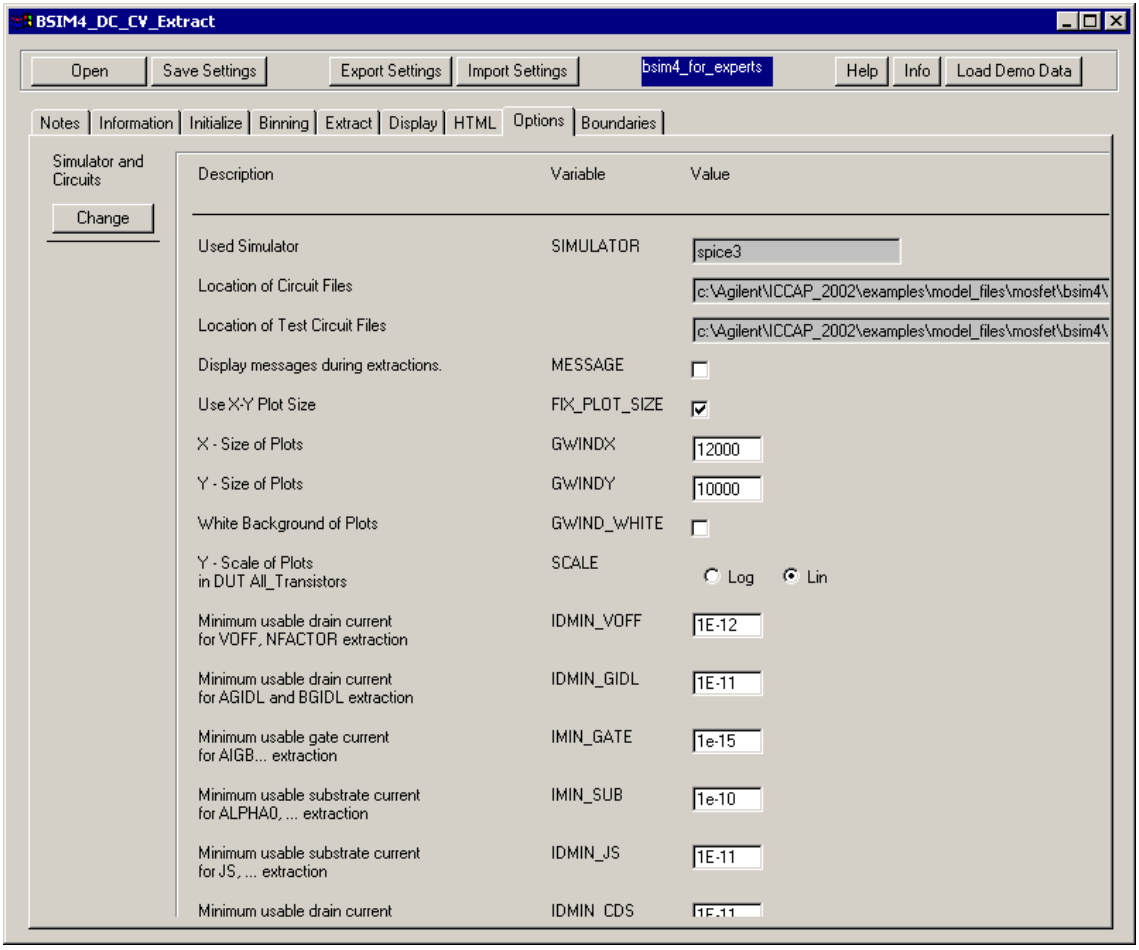
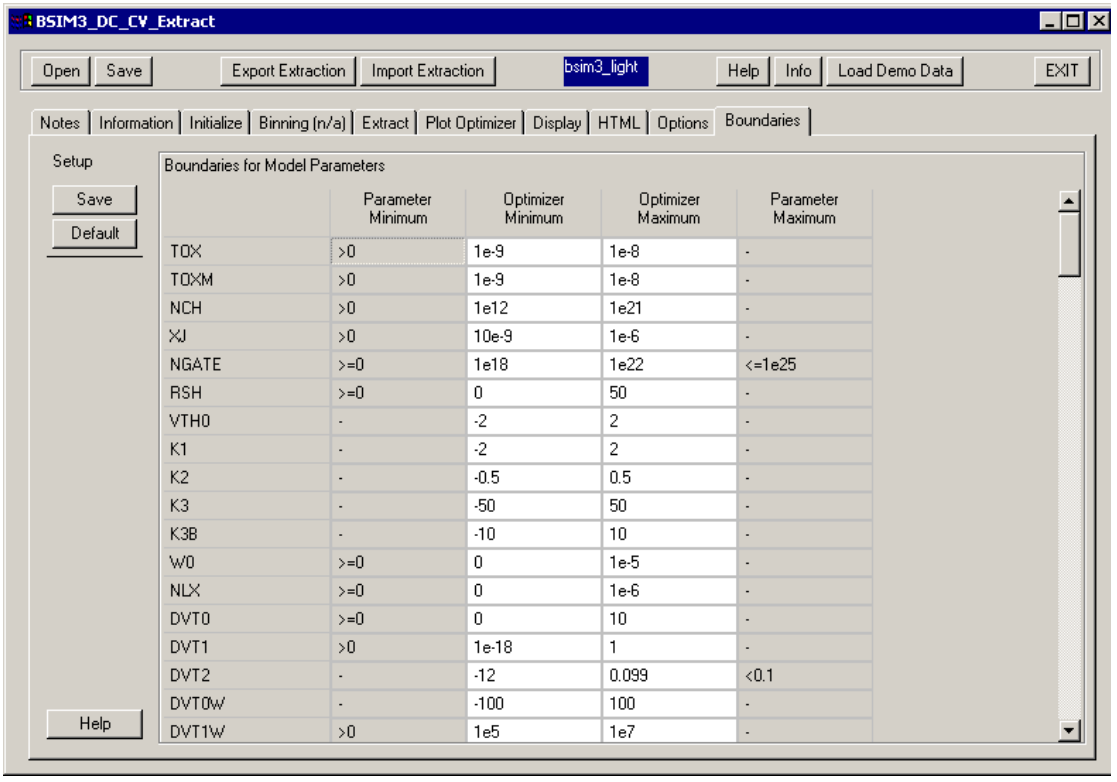


Figure 63 Options folder for the BSIM4 model

DC Boundaries

The folder *Boundaries* is intended to set optimizer boundaries for some parameters. Since the parameters differ between BSIM3 and BSIM4, there are little differences in the look of the boundaries folders.

BSIM3 — Boundaries



	Parameter Minimum	Optimizer Minimum	Optimizer Maximum	Parameter Maximum
TOX	>0	1e-9	1e-8	-
TOXM	>0	1e-9	1e-8	-
NCH	>0	1e12	1e21	-
XJ	>0	10e-9	1e-6	-
NGATE	>=0	1e18	1e22	<=1e25
RSH	>=0	0	50	-
VTH0	-	-2	2	-
K1	-	-2	2	-
K2	-	-0.5	0.5	-
K3	-	-50	50	-
K3B	-	-10	10	-
W0	>=0	0	1e-5	-
NLX	>=0	0	1e-6	-
DVT0	>=0	0	10	-
DVT1	>0	1e-18	1	-
DVT2	-	-12	0.099	<0.1
DVT0W	-	-100	100	-
DVT1W	>0	1e5	1e7	-

Figure 64 Set Optimizer Boundaries for the BSIM3 model

BSIM4 — Boundaries

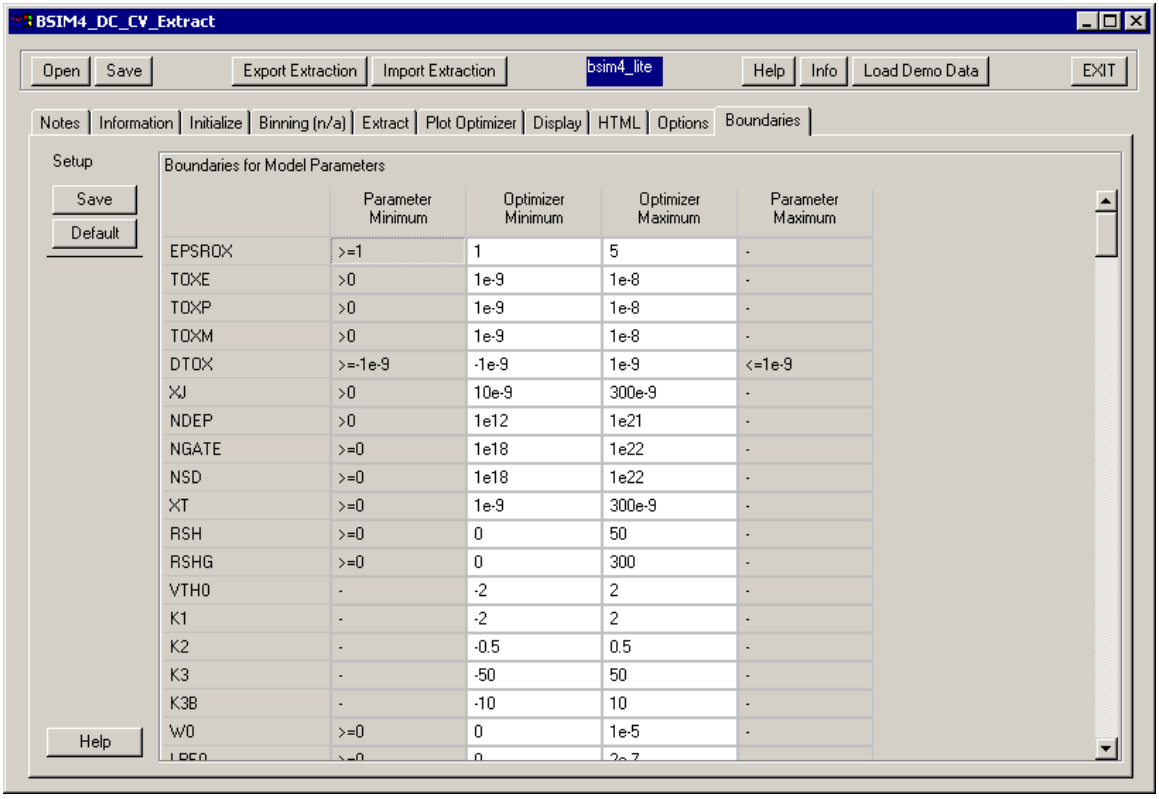


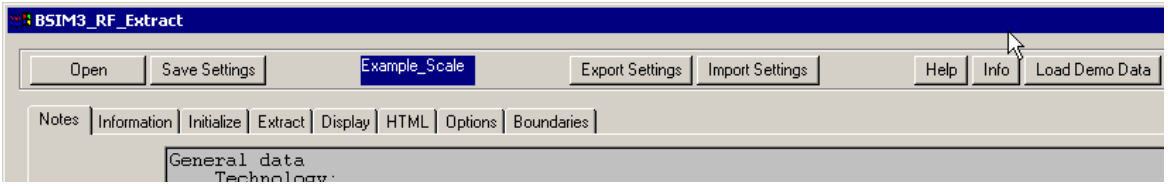
Figure 65 Set Optimizer Boundaries for the BSIM4 model

On the left side, you can see the parameter to be optimized. The following columns display the minimum for the named parameter (the parameter's reasonable physical minimum), an optimizer minimum and maximum column, followed by the parameter's maximum, if a reasonable one exists. The white fields let you enter optimizer settings fitting your process needs. You can *Save* these settings for future extractions using the *Export* button inside the *Setup* field to the left. You can restore boundaries by clicking *Default*.

Please browse the directory *$ICCAP_ROOT/examples/model_files/ mosfet/bsim4/examples/boundaries* to find typical parameter boundaries for different CMOS process generations from gate lengths of 0.8µm down to 0.13µm.

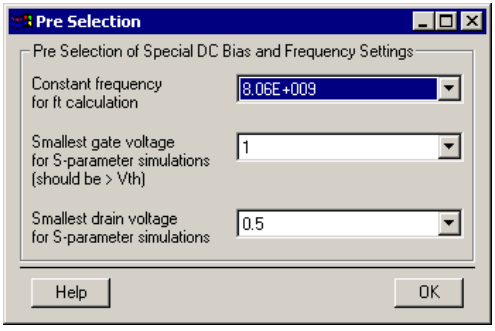
Extraction of Parameters for the RF Models

Start extraction of RF parameters for the BSIM3 or BSIM4 models by clicking the appropriate extract model to open the graphic user interface (GUI) you are already familiar with. The tasks are separated on subfolders for easy handling. Some of the folders are using the same look as in the DC Extraction part of the BSIM3 and BSIM4 Modeling Tools.



The top row buttons are described in "DC and CV Measurement of MOSFET's for the BSIM3 and BSIM4 Models" on page 77.

As soon as you click *Open*, the PreSelection dialog box opens, prompting you for some basic definitions for parameter extraction.



Within this window, select some DC and frequency settings for the extraction process.

Transit frequency f_T of a transistor is being calculated using the standard procedure of measuring the gain at a predefined frequency and extrapolating f_T from the gain-bandwidth product of one. Enter the frequency to be

used for extraction into the PreSelection window. Only frequencies defined in the measurement section using the Measurement Conditions folder are allowed. Inside this folder you've entered *Start* and *Stop Frequency* as well as *Number of Frequencies* to be measured. Frequency sweep divided by number of frequency points results in specific frequencies to be measured. Those are the frequencies you are able to select as constant frequency for f_T calculation. Be aware of the network analyzer's accuracy at lower frequencies when selecting the calculation frequency.

You can further specify the smallest gate and drain voltages to be used for S-parameter simulations. Choose the minimum gate voltage to be greater than the threshold voltage to ensure that the device is operating inside the active region. Otherwise, there will be a problem in extracting R_{out}. This resistance is very high if the transistor is turned off, resulting in large errors during extraction. Measurement is being carried out at gate and drain voltages from zero volts upward, but parameter extraction will lead to erroneous values.

RF Extract Notes

The Notes folder is described already. See "Project Notes" on page 80.

RF Extract Information

The second folder, Information, has the same look and function as the one in DC Extraction. See "DC Information" on page 139.

RF Extract Initialize

The folder Initialize is used to set initial values during extraction for process and geometric parameters as well as model flags. There are differences in initializing BSIM3 and BSIM4 models. The following figures show the initialization folders of each of the models.

BSIM3 — Initialize

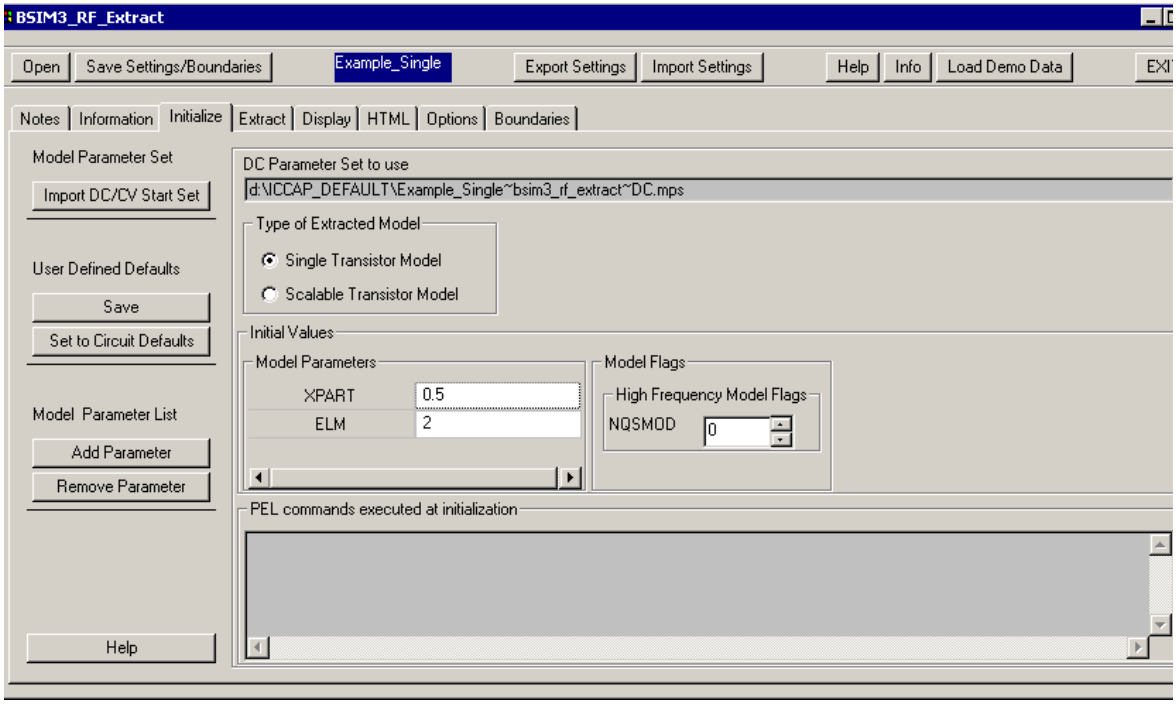


Figure 66 Initialize folder for the BSIM3 model

BSIM4 — Initialize

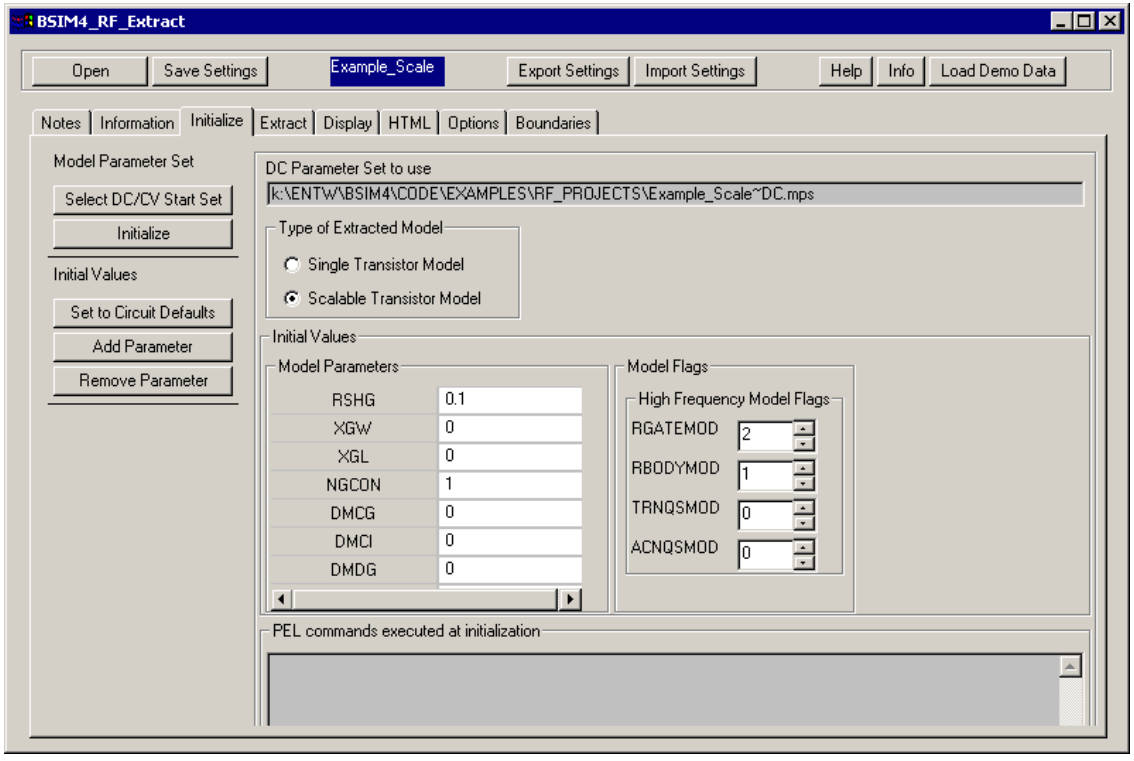


Figure 67 Initialize folder for the BSIM4 model

The Initialize folders contain sections to enter PEL commands to be executed at initialization of the extraction process.

Model Parameter Sets

To the left of the folder, you can *Import DC/CV Start Set* to be used for RF extraction. You get a list of existing *.mps files for selection. A selected *.mps file will be copied into the RF project directory. This action will set parameter values extracted from DC/CV measurements as starting points for RF extraction. Since the devices measured for RF

extraction are very compact multifinger transistors due to design requirements (and also to enhance accuracy through reducing measurement noise during network analyzer measurements), their parameters differ from the ones extracted from DC measurements. To get results consistent for the process—not only for the actual measured device—the extraction of RF relevant parameters must start with initial parameter start points to fit the S-parameters at low frequencies. Therefore, using parameters extracted during the DC extraction process are used to give start points of good accuracy for the RF extraction process.

The path and filename of the selected start set will be shown on blue background.

NOTE You cannot change directly the path and filename of the start set in the field *DC Parameter Set to use*. Instead, use the *Import DC/CV Start Set* button to the left of the folder to enter the correct path or to browse for the location of your start set.

Set parameters to circuit default values: Choosing the *Set to Circuit Default Values* button on the left side of the folder inside the *User defined Defaults* section restores the defaults.

You can *Add* or *Remove Parameters* from the *Initial Values* list in the middle of the *Initialize* folder.

Select the type of model to be extracted. There are two selections possible: *Single* or *Scalable Transistor Model*. For a detailed explanation, see “Single Transistor Model” on page 385 or “Fully scalable device” on page 386.

You can set initial parameter values manually. To do so, enter the desired values into the parameter fields provided for several parameters.

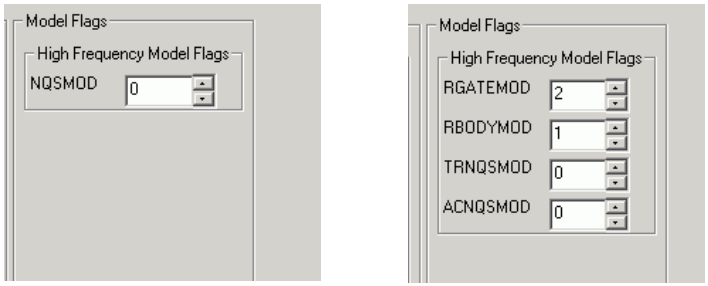


Figure 68 High Frequency Model Flag Selection for the BSIM3 (left) and for the BSIM4 (right) RF models (Part of Initialize Folder)

The BSIM3 model only uses one flag for RF modeling–the NQSMOD flag (non-quasi-static model, see "Non-Quasi-Static Model Parameters" on page 290).

Set high frequency *Model Flags* for the extraction of BSIM4 parameters by using the arrows provided to change the flag value. The process is limited to allowed flag values of the respective RF model.

The flags can have values as listed in the following table.

Table 11 High Frequency Model Flags for BSIM4

	Values	Meaning
RGATEMOD [Page 8-8]	0 (no gate resistance) 1 (constant gate resistance) 2 (variable gate resistance) 3 (two gate resistances, overlap capacitance current will not pass through intrinsic input resistance)	Gate resistance model selector
RBODYMOD [Page 8-9]	0 (no substrate resistance network) 1 (five substrate resistors are present)	Substrate resistance network model selector
TRNQSMOD [Page 8-3]	0 (charge deficit NQS model is off) 1 (charge deficit NQS model is on)	Transient Non-Quasi-Static (NQS) model selector

Table 11 High Frequency Model Flags for BSIM4 (continued)

	Values	Meaning
ACNQSMOD [Page 8-5]	0 (small signal AC charge deficit NQS model is off) 1 (small signal AC charge deficit NQS model is on)	AC small-signal Non-Quasi-Static model selector

Note [Page 8-X] refers to the page numbers of the BSIM4.3.0 Manual from UC Berkeley [1]

RF Extract

Within this folder, you define the extraction process for the parameters of the devices. There is a standard extraction flow implemented, but you can change this flow if you find another one suits your needs better than the default flow.

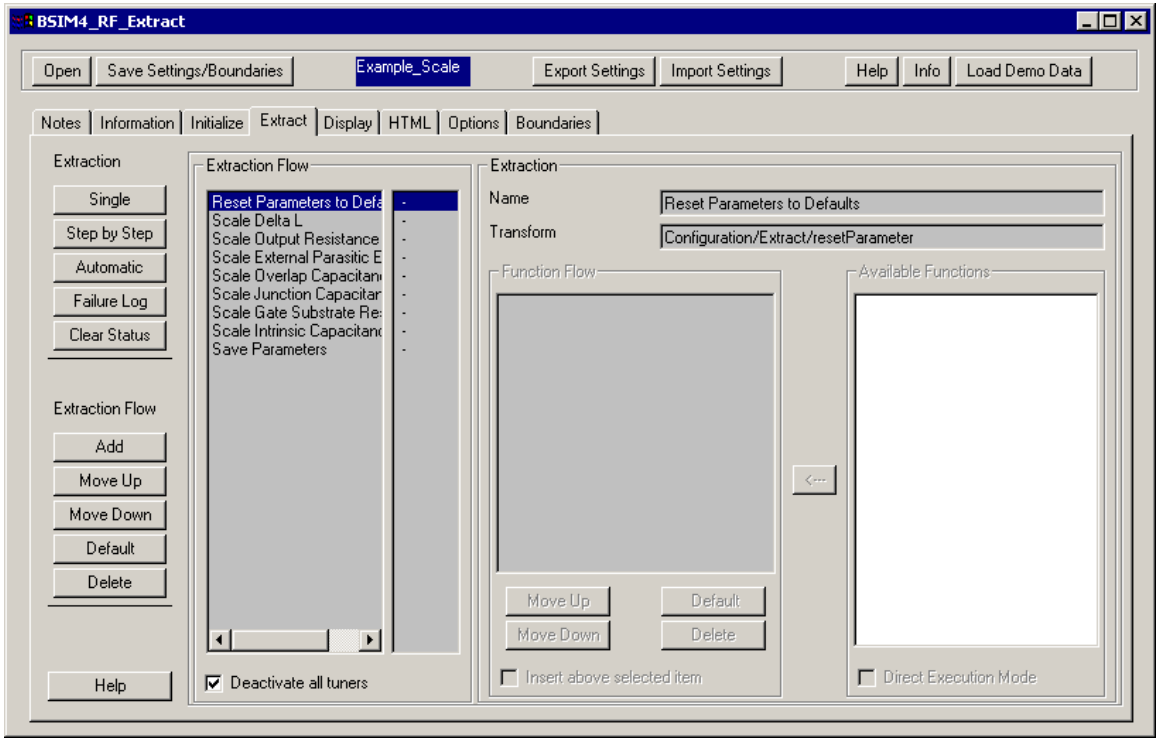


Figure 69 Extract Folder

- To extract parameters from one flow: Select the desired flow under the *Extraction Flow* section of this folder and choose the *Single* button. Only the selected extraction will be performed. The status of extraction is visible in the status field. This field shows a '-' if extraction of this parameters is not completed yet or 'done' if the parameters from this step are extracted.
- To go through the extraction process one step at a time, highlight the step then choose *Step by Step*. A dialog box may appear, prompting you for input.

- To automatically extract all parameters using the extraction flows listed under the *Extraction Flow* section: Choose *Automatic*. The programmed extraction flow will be extracting all parameters defined in the active extraction flow.
- All warnings and errors during the extraction process are written to the failure log, which is opened using the *Failure Log* button.

If you would like to clear the status of extraction, use *Clear Status*.

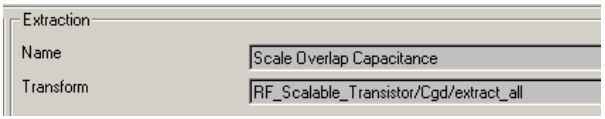
NOTE Already extracted parameters are reset to defaults!

- You can add steps to the extraction flow by clicking the *Add* button on the left side of the folder under the section Extraction Flow. You will be prompted for an extraction to add. Select the desired extraction and choose *Add* on the Add Extraction folder.
- Change the flow of extraction by using the *Move Up* or *Move Down* buttons to move a selected extraction routine one step up or down.
- The *Default* button restores the order of parameter extraction as it was in the beginning of a project.
- To delete a step: Choose the *Delete* button.

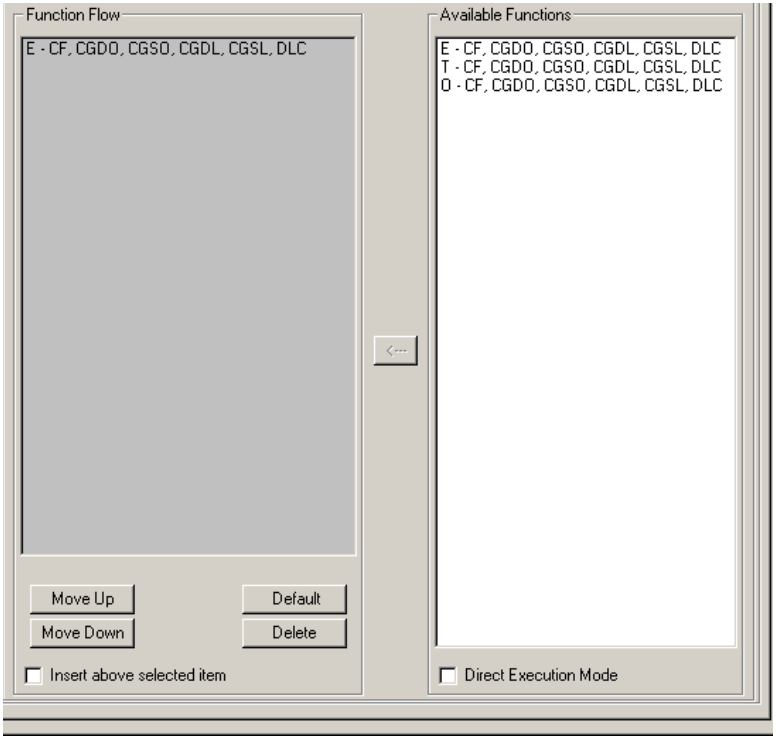
NOTE You cannot delete the first (Reset Parameters) and the last (Save Parameters) step inside an extraction flow.

- To export the extracted parameters: The step *Save Parameters* inside the *Extraction Flow* informs you of the path and name for the saved .mps or .lib file.

On the right side of the Extract folder, you will find a field named *Extraction*.



This field shows the name of the extraction as well as name and path of the transform used in this extraction step. There is a field, *Function Flow*, which is used to set the flow of extraction steps. Select the desired function out of the list found under *Available Functions* by selecting the function and clicking the arrow in between the *Function Flow* and the *Available Functions* fields. Arranging the functions inside the function flow is done using the buttons provided below the *Function Flow* field.


Function Flow
E - CF, CGDO, CGSO, CGDL, CGSL, DLC
Move Up
Move Down
Default
Delete
Insert above selected item
Available Functions
E - CF, CGDO, CGSO, CGDL, CGSL, DLC
T - CF, CGDO, CGSO, CGDL, CGSL, DLC
O - CF, CGDO, CGSO, CGDL, CGSL, DLC
Direct Execution Mode

Under Available Functions you will find in the example above three functions for this function flow. The first function is an extraction step (E), the second one uses the tuner (T), the third one uses the optimizer (O).

RF Extract Display

Within this folder, you will find fields to select plots for display.

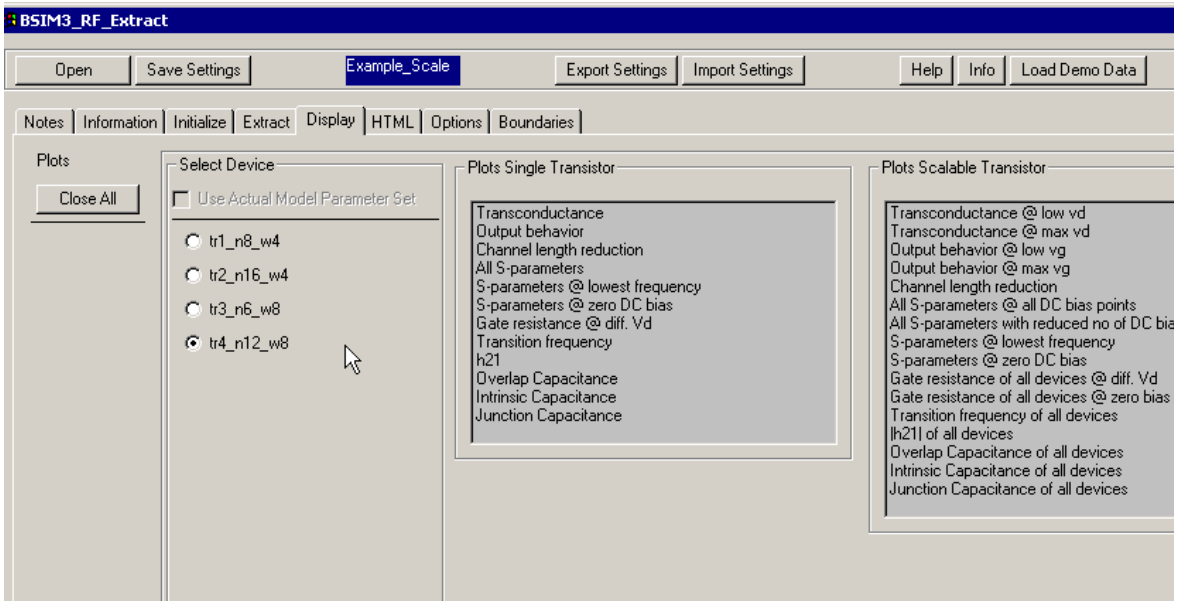


Figure 70 Display folder

There are three fields to select models for display on this folder:

Select Device: *Use Actual Model Parameter Set* or choose one of the buttons to the left of the transistors name for plots to be displayed.

Open plots for the selected transistor by clicking the name of the plot inside the *Plots Single Transistor* field on the folder.

Open plots for scalable transistor models by clicking the name of the desired plots inside the *Plots Scalable Transistor* field on this folder.

NOTE You can only choose a plot for the extracted model that was selected in the field *Type of Extracted Model* on the *Initialize* folder. Plots inside the other fields are not selectable.

Each plot is opened in a new window. Close all windows by using the *Close All* button on the right side of the Display folder or close single windows using the Close icon inside the appropriate window.

RF Extract HTML

This folder helps you to prepare a report file in HTML format to be displayed using an internet browser.

Since this folder is the same as in the DC Extraction section, see "DC HTML" on page 165 for details.

RF Extract Options

This folder is used to set variables for the extraction process and options for plots to be displayed.

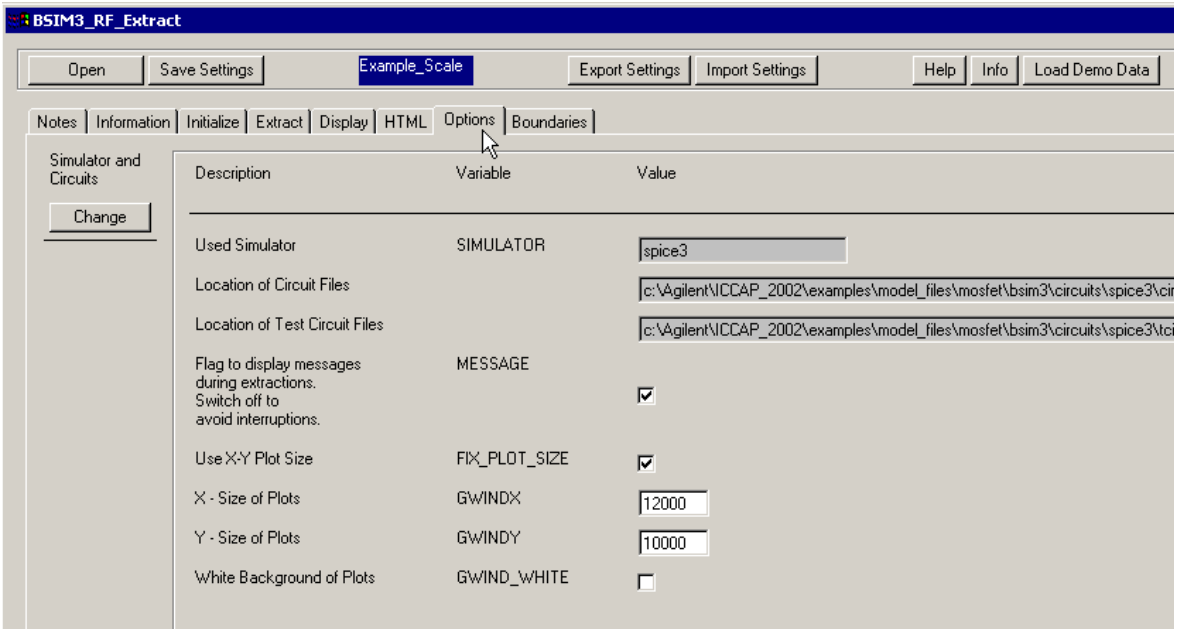


Figure 71 Options Folder

The Simulator variable field and the paths to circuit and test circuit files have a blue background. You cannot change those field entries directly. Instead, you have to choose the Simulator and Circuits *Change* button to set another target simulator as well as paths to circuit and test circuit files. If you use your own circuit files, it is recommended to copy the entire *examples* directory into a directory where you have write access and set the path according to your situation. You will find the path to the examples directory on the options folder! If you are satisfied with the default settings or to use as a starting point, just leave the path entries as provided.

There is a field to enter the printer command for printing the plots (see the notes on printing in the section "DC and CV Measurement of MOSFET's for the BSIM3 and BSIM4 Models" on page 77).

If you want to change the plot window size, choose *Use X-Y Plot Size: Fixed_Plot_Size* and enter the desired X and Y size into the respective fields. You are able to change the background color of the plot window from black to white by activating the field: *White background of plots*.

Circuit Files

The circuit files are located in: *~ICCAP_ROOT/example/model_files/mosfet/bsim3(or bsim4)/circuits*

You will find subdirectories below the circuits directory for each supported simulator (hpeesofsim, hspice, spectre, spice3). Each directory contains a circuit (cir) as well as a test circuit (tci) directory, which contain the circuit files using the appropriate simulator syntax.

RF Extract Boundaries

The Boundaries folder is the same as the one in DC Extraction. For details, see "DC Boundaries" on page 172.

4
BSIM3v3 Characterization

The BSIM3 Model 197
The Unified I-V Model of BSIM3v3 200
Capacitance Model 238
High Frequency Behavior 255
Temperature Dependence 271
Noise Model 279
SPICE Model Parameters 282
Test structures for Deep Submicron CMOS Processes 293
Extraction of Model Parameters 311
Binning of Model Parameters 318
Importing older version BSIM3v3 Files 328
References 332

This Chapter explains the theoretical background of the BSIM3 model. Using the Modeling Package is described in Chapter 3, “Using the BSIM Modeling Packages”.

What's new inside the BSIM3 Modeling Package:

This section lists the enhancements and changes made to the Modeling Package for each revision since IC-CAP 2002. They are listed in reverse order so that the new version is on top, followed by changes made in former versions.

New features in the BSIM3 Modeling Package, Rev. IC-CAP 2004, spring 2005

1.) BSIM3_DC_CV_Measure

The time to load a new project has been dramatically reduced (also in BSIM3_DC_CV_Extract, BSIM3_RF_Measure, BSIM3_RF_Extract).

List sweeps are now supported.

2.) BSIM3_DC_CV_Extract

The extraction flow has been enhanced to store and retrieve complete extraction scenarios including intermediate results and boundary settings.

The usability of the plot optimizer inside the BSIM3 Package has been enhanced and user configured plot optimizers can be easily integrated into the extraction flow.

3.) BSIM3_RF_Measure

New scheme to define de-embedding structures.

New features in the BSIM3 Modeling Package, Rev. IC-CAP 2004, January 2004

1.) General

The Graphic User Interface from BSIM4 has been adopted. One of the main advantages of this concept is that the measured data can be used by BSIM3 and BSIM4 Modeling Packages for parameter extraction!!!

The BSIM3 Modeling Package can now generate model cards and scalable RF models for the following simulators:

Spice3 (delivered with IC-CAP)

Advanced Design System

Hspice

Spectre

The documentation was totally reworked to account for the common user interface with the BSIM4 Modeling Package and similar upcoming modeling products.

In addition, a detailed description of all the files of the Modeling Package is given. All temperatures in the setup and documentation are now given in [K] instead of [degree C].

Currently, the supported model is BSIM3v3.2.4, released on Dec.21st, 2001.

2.) BSIM3_DC_CV_Measure

The Keithley switching matrix models K707 and K708a are now supported.

The maximum compliance values can now be defined together with the other measurement settings.

Three new functions are implemented to drive the BSIM3_DC_CV_Measure module from a wafer prober control macro. An example for such a control macro can be found in *'.../examples/model_files/mosfet/BSIM3/examples/waferprober/prober_control.mdl'*

3.) BSIM3_DC_CV_Extract

A complete new extraction flow is implemented. A certain extraction group (e.g., 'Basic VTH, Mobility') can be invoked several times with different configurations.

Moreover, the flow inside an extraction group can be specified in any desired order.

This gives the highest available flexibility for adopting any parameter extraction to a certain process.

Automatic generation of binned model files is now supported. A new folder *Binning* in the *BSIM3_DC_CV_Extract* module allows the specification of the binning areas as well as extended binning. Final circuits are generated for Hspice, Spectre, and ADS.

Generation of HTML files has been enhanced to include a navigation tree through all results. In addition, all measured data at each temperature for each device is compared with the simulated results.

The new IC-CAP feature *Plot Optimizer* is supported by a user friendly configuration of the devices and setups for a final fine tuning approach.

A new function is implemented to extract multiple projects in batch mode. This can be very useful for statistical modeling, where a large number of model parameter sets must be generated for the same type of devices but from different measured test chips. Please see the macro '*Example_Wafer_Extraction*' in the *BSIM3_DC_CV_Extract.mdl* file.

Parameter extractions have been steadily enhanced due to user's feedback.

4.) BSIM3_RF_Measure

No changes made.

5) BSIM3_RF_Extract

A complete new extraction flow is implemented. Please see 3.) *BSIM3_DC_CV_Extract* for more details.

The automatic generation of HTML files has been enhanced to include a navigation tree through all results.

6) Documentation

The documentation was totally reworked to account for the common user interface with the BSIM4 Modeling Package and similar upcoming modeling products.

In addition, a detailed description of all files of the Modeling Packages is included.

New features in the BSIM3 Modeling Package, Rev. IC-CAP 2002, March 2003

1.) General

This is an update to the already existing BSIM3v3 Modeling Package in IC-CAP.

The complete user interface and data structure was modified and reworked to have the same style as the existing BSIM4 Modeling Package.

One of the main advantages of this concept is the usage of measured data by the BSIM3 Modeling Package as well as the BSIM4 Modeling Package for parameter extraction

Please note, for compatibility reasons the old BSIM3v3 files can still be accessed in the *$ICCAP_ROOT/examples/model_files/mosfet/bsim3v3* directory.

The new style files are located in the directory:
`$ICCAP_ROOT/examples/model_files/mosfet/bsim3`

Don't get confused by the missing version information of the "bsim3" term. The new style files don’t use the version information any more.

2.) BSIM3_DC_CV_Measure

A feature "Import BSIM3v3" was added to reuse data in the file format of the former BSIM3v3 Modeling Package.

The measured data of the new BSIM3 Modeling Package is now in a format that can be used for the generation of BSIM3 *and* BSIM4 models

3.) BSIM3_DC_CV_Extract

The existing extraction functions have been ported to the new style user interface.

A new, more user friendly HTML report can be generated, which allows a comparison of measured and simulated data for each device. In addition, the report can be easily included in a word processing program.

4.) BSIM3_RF_Measure

This module measures all data which is necessary for the generation of RF models. The data is compatible with the BSIM4_RF_Measure module and can also be used for the generation of BSIM4 RF models.

5) BSIM3_RF_Extract

A new, fully scalable subcircuit model for the BSIM3 RF behavior was added. The user can now select whether he wants to create a single device model (one model for each test device) or a fully scalable model that covers all available test devices.

6) BSIM3_Tutorial

These are the well known files for learning more about the BSIM3 model itself.

7) Documentation

The *Help* buttons are still linked to the BSIM4 *Online Help*. This is OK, because the usage of the BSIM3 Modeling Package and the BSIM4 Modeling Package is identical.

For more information about the BSIM3 model itself, please refer to this chapter.

A reworked version of the documentation will be included in the IC-CAP 2004 release.

The BSIM3 Model

The BSIM3 model (BSIM = Berkeley Short channel Insulated gate field effect transistor Model) was published by the University of California at Berkeley in July 1993. BSIM3 is a public model and is intended to simulate analog and digital circuits that consist of deep submicron MOS devices down to channel lengths of 0.18 micron.

BSIM3 is a physical model with built-in dependencies of important device dimensions and process parameters like the channel length and width, the gate oxide thickness, substrate doping concentration and LDD structures. Due to its physical nature and its built-in geometry dependence, the prediction of device behavior of advanced devices based on the parameters of the existing process is possible. As a further improvement, one set of model parameters covers the whole range of channel lengths and channel widths of a certain process that can be used in circuit designs. Due to the physical meaning of many model parameters, the BSIM3 model is the ideal basis for the statistical analysis of process fluctuations.

BSIM3 can model the following physical effects of modern submicron MOS transistors:

- Threshold Voltage
  - Vertical and lateral non-uniform doping
  - Short channel effects
  - Narrow channel effects
- Mobility
  - Mobility reduction due to vertical fields
- Carrier Velocity Saturation
- Drain Current
  - Bulk charge effect
  - Subthreshold conduction

  - Source/drain parasitic resistance
- Bulk Current
- Output Resistance
  - Drain induced barrier lowering (DIBL)
  - Channel length modulation (CLM)
  - Substrate current induced body effect (SCBE)
- Short channel capacitance model
- Temperature dependence of the device behavior

For a detailed description of these features, refer to the BSIM3 manual from Berkeley University. You can order this manual from Berkeley or you can get it over the Internet. See "References" on page 332 for details.

The BSIM3v3 Modeling Package provides a complete extraction strategy for the model parameters of the BSIM3v3.2.4 model. The extraction routines are based on the BSIM3v3.2.4 device equations to ensure that the extracted model parameters represent as good as possible the original physical meaning. Therefore, no or only a minimum of optimization is needed to get a good fit between measured and simulated device behavior.

The routines of this release refer to version 3.2.4 of the BSIM3 model that was released by University of California at Berkeley in December 2001.

Versions of the BSIM3 Model

University of California at Berkeley released four versions of its BSIM3 model. The first three versions have differences in some model parameters, and the model parameter sets are not compatible.

The following example of the parameter UC, which is a part of the mobility reduction, demonstrates the problem:

In BSIM3v2, the effective mobility μeff was calculated according to the following formula:

$$\mu_{eff} = \frac{\mu_o}{1 + U_a((V_{gs} + V_{th})/T_{ox}) + U_b((V_{gs} + V_{th})/T_{ox})^2 + U_c V_{bs}}$$

In BSIM3v3.2.2, the formula changed to:

$$\mu_{eff} = \frac{\mu_o}{1 + (U_a + U_c V_{bseff})((V_{gsteff} + 2V_{th})/T_{ox}) + U_b((V_{gsteff} + 2V_{th})/T_{ox})^2}$$

It can easily be recognized, that UC has quite different values in both equations.

That means, if BSIM3v2 is implemented in the simulator and the parameter is extracted for BSIM3v3.2.2, the simulation will give catastrophic results (in the case of UC).

Therefore, you must be sure that you use the same version of BSIM3 in both your simulator and your extraction tool.

The latest release, BSIM3v3.2.4 is a minor change to BSIM3v3.2.2 with only a few bug fixes. The model equations used are the same in those versions.

The Unified I-V Model of BSIM3v3

For a complete summary of all equations of the BSIM3v3.2.4 model, please refer to the original documentation from University of California at Berkeley (see "References" on page 332 to order this paper). The main equations of the BSIM3v3.2.4 model are shown together with a graphical representation for a better understanding of the model.

Please use the models *BSIM3_DC_Tutorial.mdl*, *BSIM3_CV_Tutorial.mdl*, *BSIM3_AC_Noise_Tutorial.mdl*, or *BSIM3_Temp_Tutorial.mdl* provided with the BSIM3 Modeling Package to visualize most of the model parameters influences onto the device diagrams. Load the file into the IC-CAP GUI to see how certain parameters affect the behavior of a deep submicron MOS transistor.

Threshold Voltage

The threshold voltage is one of the most important parameters of deep submicron MOS transistors and is affected by many different effects when the devices are scaled down into the region of 0.1 microns. The complete equation of the threshold voltage in BSIM3v3.2.4 is given below.

(1)

$$V_{th} = V_{Tideal} + \Delta V_{th(1)} + \Delta V_{th(2)} - \Delta V_{th(3)} - \Delta V_{th(4)} + \Delta V_{th(5)} - \Delta V_{th(6)}$$

The different parts of this complex equation are expressed by the following sub-equations in more detail:

(2)

$$
\begin{aligned}
V_{th} = {} & V_{th0} - K_1\sqrt{\Phi_s} \\
& + K_1\frac{T_{ox}}{T_{oxm}}\sqrt{\Phi_s - V_{bseff}} - K_2\left(\frac{T_{ox}}{T_{oxm}}V_{bseff}\right) \\
& + K_1\frac{T_{ox}}{T_{oxm}}\left(\sqrt{\left(1+\frac{Nlx}{L_{eff}}\right)} - 1\right)\sqrt{\Phi_s} \\
& - D_{VT0}\left[e^{\left(-D_{VT1}\frac{L_{eff}}{2l_t}\right)} + 2e^{\left(-D_{VT1}\frac{L_{eff}}{l_t}\right)}\right](V_{bi} - \Phi_s) \\
& - D_{VT0w}\left[e^{\left(-D_{VT1w}\frac{W_{eff}L_{eff}}{2l_{tw}}\right)} + 2e^{\left(-D_{VT1w}\frac{W_{eff}L_{eff}}{l_{tw}}\right)}\right](V_{bi} - \Phi_s) \\
& + (K_3 + K_{3b}V_{bseff})\frac{T_{ox}}{(W_{eff} + W_0)}\Phi_s \\
& - \left[e^{\left(-D_{sub}\frac{L_{eff}}{2l_{t0}}\right)} + 2e^{\left(-D_{sub}\frac{L_{eff}}{l_{t0}}\right)}\right](E_{ta0} + E_{tab}V_{bseff})V_{ds}
\end{aligned}
$$

Ideal Threshold Voltage

The basic equation of the threshold voltage is:

$$V_{Tideal} = V_{th0} = V_{FB} + \Phi_s + K_1\sqrt{\Phi_s} \quad \textbf{(3)}$$

$$\Phi_s = 2V_{tm0}\, ln\left(\frac{N_{ch}}{n_{i0}}\right) \text{ at } T = T_{nom} \quad \textbf{(4)}$$

$$V_{tm0} = \frac{k_B T_{nom}}{q} \quad \textbf{(5)}$$

where:

$V_{thideal}$ = ideal threshold voltage

V_{FB} = flatband voltage

Φ_s = surface potential

n_i = $1.45 \cdot 10^{10} (T_{nom}/300.15)^{1.5} (21.5566 - E_{g0}/2V_{tmo})$

E_g0 = $1.16 - 7.02 \cdot 10^{-4} T_{nom}^2/(T_{nom} + 1108)$

This equation had been implemented into the first MOS simulation models assuming long and wide channels and uniform substrate doping. The following sections describe the effects that overlay this basic equation.

Non-Uniform Vertical Channel Doping

The substrate doping concentration N is not constant in the vertical direction of the channel, as shown in Figure 72.

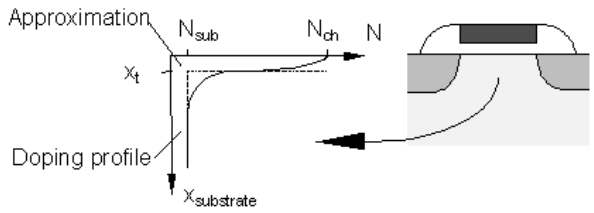


Figure 72 Vertical Doping Profile in the Channel

It is usually higher near the silicon to silicon dioxide interface than deeper in the substrate. This higher doping concentration is used to adjust the threshold voltage of the device. The distribution of impurity atoms inside the substrate is approximately a half Gaussian distribution, which can be approximated by a step function with NCH for the peak concentration in the channel near the Si-SiO_2 interface and Nsub in the deep bulk. XT is the depth where the approximation of the implant profile switches from NCH to NSUB. The non-uniform vertical channel doping affects the threshold voltage when a bulk source voltage is applied to the device and is represented here as the part $\Delta V_{th(1)}$ of the overall threshold voltage.

$$\Delta V_{th(1)} = K_1 \frac{T_{ox}}{T_{oxm}} \sqrt{\Phi_s - V_{bseff}} - K_2 \frac{T_{ox}}{T_{oxm}} V_{bseff} \quad (6)$$

$$K_1 = \gamma_2 - 2K_2 \sqrt{\Phi_s - V_{bm}} \quad (7)$$

$$K_2 = \frac{(\gamma_1 - \gamma_2)(\sqrt{\Phi_s - V_{bx}} - \sqrt{\Phi_s})}{2\sqrt{\Phi_s}(\sqrt{\Phi_s - V_{bx}} - \sqrt{\Phi_s}) + V_{bx}} \quad (8)$$

$$\gamma_1 = \frac{\sqrt{2q\varepsilon_{si}N_{ch}}}{C_{ox}} \quad (9)$$

$$\gamma_2 = \frac{\sqrt{2q\varepsilon_{si}N_{sub}}}{C_{ox}} \quad (10)$$

where:

V_{bx} = substrate bias voltage when the depletion width equals

$$X_t = \Phi_s - \frac{qN_{ch}X_t^2}{2\varepsilon_{si}}$$

V_{bm} = maximum substrate bias voltage

T_{oxm} = gate oxide thickness at which parameters are extracted

T_{ox} = default value of T_{oxm}

V_{bseff} = $V_{bc} + 0.5\left[V_{bs} - V_{bc} - \delta_1 + \sqrt{(V_{bs} - V_{bc} - \delta_1)^2 - 4\delta_1 V_{bc}}\right]$

δ_1 = 0.001V

V_{bc} = $0.9\left(\Phi_s - \frac{K_1^2}{4K_2^2}\right)$

In BSIM3, either the model parameters K1 and K2 or NCH, NSUB, VBM or XT can be used to model this effect. Figure 73 shows the threshold voltage V_{th} as a function of the applied bulk voltage for a transistor with a large channel length and a wide channel width (LARGE).

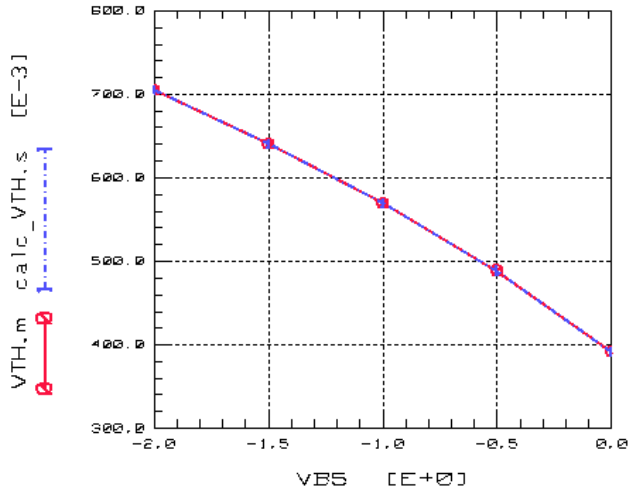


Figure 73 Threshold Voltage V_{th} as a Function of V_{bs}

Non-Uniform Lateral Channel Doping

The doping concentration N_{ds} near the drain and the source is higher than the concentration N_a in the middle of the channel. This is referred to as lateral non-uniform doping concentration and is shown in Figure 74.

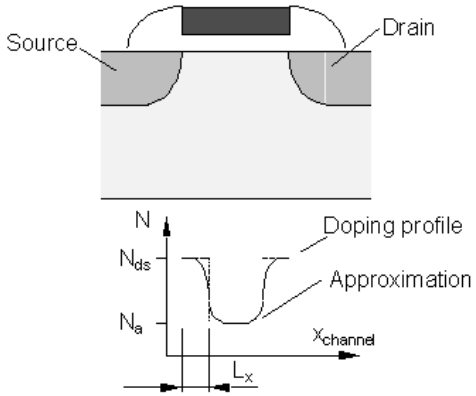


Figure 74 Lateral Doping Profile in the Channel

As the channel length becomes shorter, the lateral non-uniform doping will cause the threshold voltage to increase strongly because the average doping concentration in the channel becomes higher. This part of the threshold voltage is modeled with the parameter Nl_x and is represented by $\Delta V_{th(2)}$ as a part of the overall threshold voltage.

$$\Delta V_{th(2)} = K_1 \frac{T_{ox}}{T_{oxm}} \left(\sqrt{\left(1 + \frac{Nl_x}{L_{eff}}\right)} - 1 \right) \sqrt{\Phi_s} \qquad \textbf{(11)}$$

where:

$$Nl_x = 2L_x(N_{ds} - N_a)/N_a$$

Figure 75 shows the influence of the non-uniform lateral doping on the threshold voltage as a function of gate length.

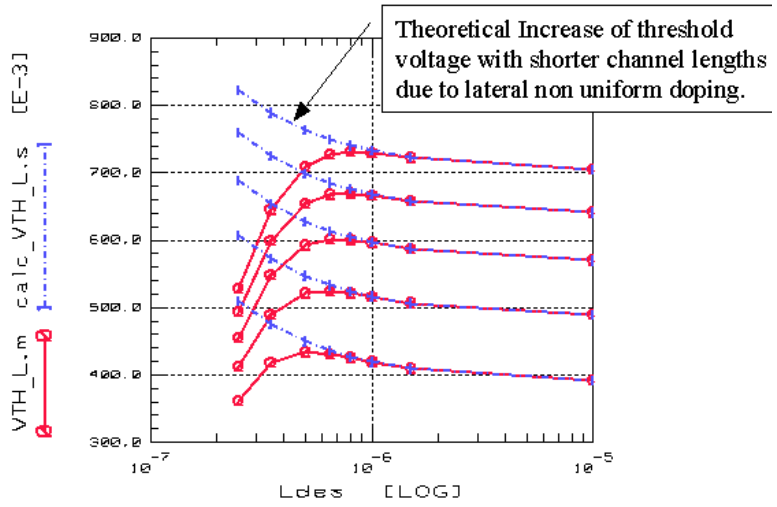


Figure 75 Threshold Voltage as a Function of Gate Length Due to Lateral Non-Uniform Doping

You can distinguish between the theoretical trace following Equation 11 and the real world ones with the short channel effect described in the next section.

Short Channel Effect

The threshold voltage of a long channel device is independent of the channel length and the drain voltage as it is shown in the equation of the ideal threshold voltage. The decreasing of device dimensions causes the so-called short-channel effects: threshold voltage roll-off and degradation of the subthreshold slope, that in turn increases the off-current level and power dissipation. The threshold voltage then depends on geometrical parameters like the effective channel length and the shape of the source-bulk and drain-bulk junctions. These device dimensions have a strong influence on the surface potential along the channel. A shallow junction with a weak lateral spread is desirable for the control of short-channel effects while the source and drain resistance must be kept as low as possible. However, a trade-off between the search for very shallow junctions and

the degradation of the maximum achievable current through the parasitic resistance of low doped drain regions must be found.

Those effects can be shown in device simulators, where drift, diffusion, and additionally the hot electron behavior can be simulated. The following equations are responsible for the modeling of the short channel effect part $\Delta V_{th(3)}$ in the BSIM3 model:

$$\Delta V_{th(3)} = D_{VT0}\left[e^{\left(-D_{VT1}\frac{L_{eff}}{2l_t}\right)} + 2e^{\left(-D_{VT1}\frac{L_{eff}}{l_t}\right)}\right](V_{bi}-\Phi_s) \quad \textbf{(12)}$$

$$l_r = \sqrt{\frac{\varepsilon_{si}T_{ox}X_{dep}}{\varepsilon_{sio2}}}(1+D_{VT2}V_{bseff}) \quad \textbf{(13)}$$

$$X_{dep} = \sqrt{\frac{2\varepsilon_{si}(\Phi_s - V_{bseff})}{qN_{ch}}} \quad \textbf{(14)}$$

where:

V_{bi} built-in voltage of the PN junction between the source/drain and the substrate

$$= \frac{K_B T}{q}\ln\left(\frac{N_{ch}N_d}{n_i^2}\right)$$

N_d = source/drain doping concentration (or in the LDD regions) if they exist

DVT0, DVT1, DVT2 are parameters used to make the model fit different technologies

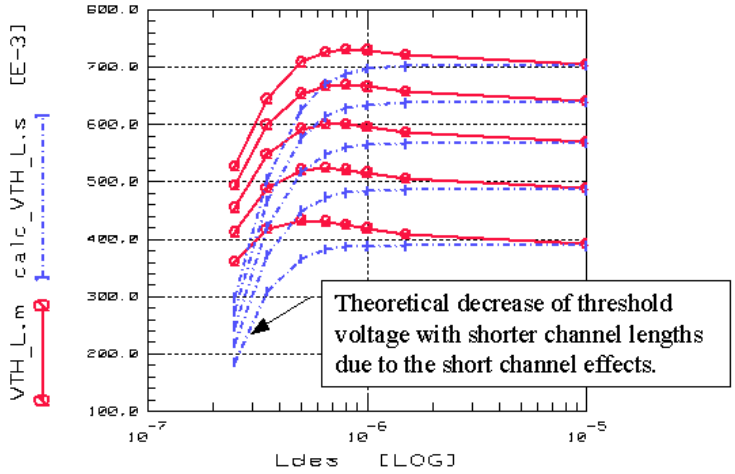


Figure 76 Influence of Short Channel Effects on the Threshold Voltage

For short channel lengths together with small channel widths, the following additional expression $\Delta V_{th(4)}$ is needed to formulate the threshold voltage:

$$\Delta V_{th(4)} = D_{VT0w}\left[e^{\left(-D_{VT1w}\frac{W_{eff}L_{eff}}{2l_{tw}}\right)} + 2e^{\left(-D_{VT1w}\frac{W_{eff}L_{eff}}{l_{tw}}\right)}\right](V_{bi}-\Phi_s) \quad \textbf{(15)}$$

where:

$$l_{tw} = \sqrt{\frac{\varepsilon_{si}T_{ox}X_{dep}}{\varepsilon_{sio2}}}(1+D_{VT2W}V_{bseff})$$

Narrow Channel Effect

All the effects on the threshold voltage are based on the non-uniformity along the channel length. Regarding the channel width, the depletion region is always larger due to the existence of fringing fields at the side of the channel. This effect becomes very substantial as the channel width

decreases and the depletion region underneath the fringing field becomes comparable to the depletion layer formed from the vertical field. This additional depletion region results in an increase of the threshold voltage with smaller channel widths, which is expressed by $\Delta V_{th(5)}$.

$$\Delta V_{th(5)} = (K_3 + K_{3b} V_{bseff}) \frac{T_{ox}}{(W_{eff} + W_0)} \Phi_s \quad \textbf{(16)}$$

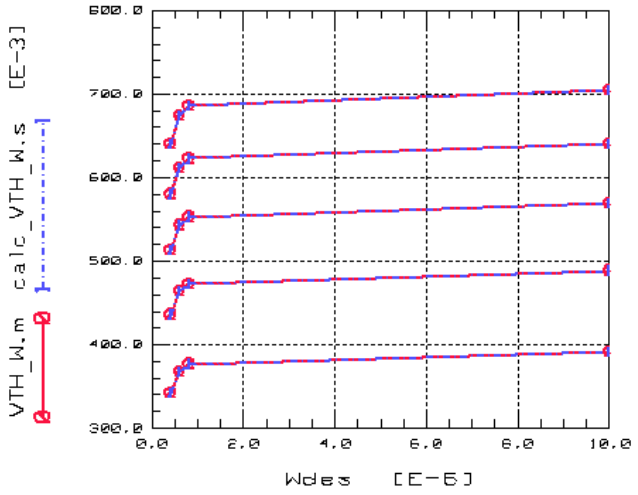


Figure 77 Influence of Narrow Channel Effects on the Threshold Voltage

Threshold Voltage Reduction Through DIBL

The effect of the drain induced barrier lowering (DIBL) will be explained later. BSIM3 uses the following equation to model the DIBL effect in the threshold voltage:

$$\Delta V_{th(6)} = \left[e^{\left(-D_{sub}\frac{L_{eff}}{2l_{t0}}\right)} + 2e^{\left(-D_{sub}\frac{L_{eff}}{l_{t0}}\right)} \right] (E_{ta0} + E_{tab} V_{bseff}) V_{ds} \quad (17)$$

$$l_{t0} = \sqrt{\frac{\varepsilon_{si} T_{ox} X_{dep}}{\varepsilon_{sio2}}} \quad (18)$$

Carrier Mobility Reduction

BSIM3v3 provides 3 different equations for the modeling of the mobility reduction. They can be selected by the flag MOBMOD.

MOBMOD=1:

$$\mu_{eff} = \frac{\mu_o}{1 + (U_a + U_c V_{bseff})((V_{gsteff} + 2V_{th})/T_{ox}) + U_b((V_{gsteff} + 2V_{th})/T_{ox})^2} \quad (19)$$

MOBMOD=2:

$$\mu_{eff} = \frac{\mu_0}{1 + (U_a + U_c V_{bseff})(V_{gsteff}/T_{ox}) + U_b(V_{gsteff}/T_{ox})^2} \quad (20)$$

MOBMOD=3:

$$\mu_{eff} = \frac{\mu_0}{1 + \left[U_a(V_{gsteff} + 2V_{th})/T_{ox} + U_b((V_{gsteff} + 2V_{th})/T_{ox})^2\right](1 + U_c V_{bseff})} \quad (21)$$

The influence of the mobility reduction parameters is demonstrated in Figure 78 where the simulated drain current with and without mobility reduction is shown.

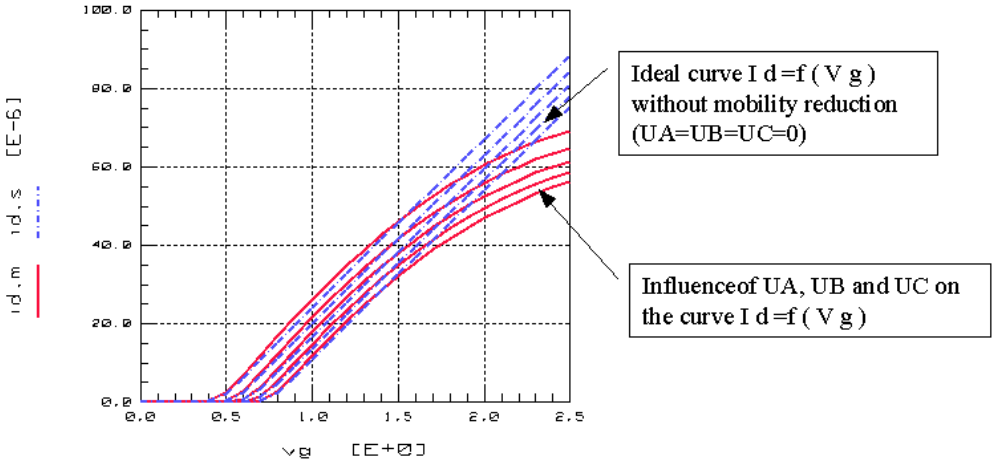


Figure 78 Influence of Mobility Reduction

Figure 79 shows the effective mobility as a function of gate voltage and bulk-source voltage.

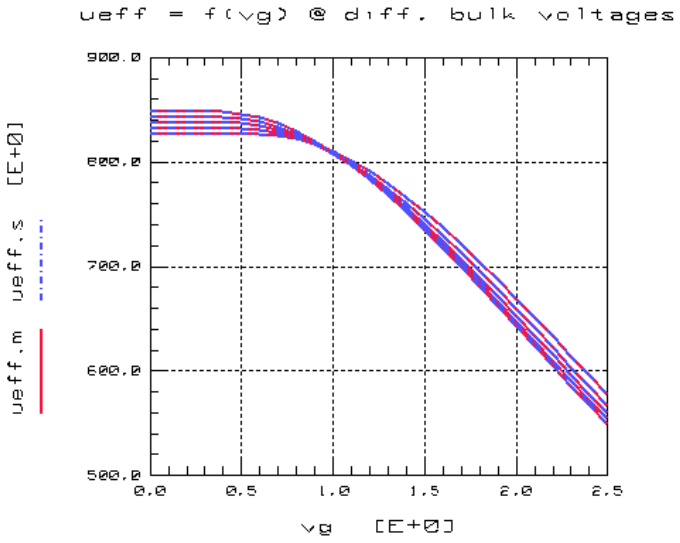


Figure 79 Effective Mobility$_{\mu eff}$ as a Function of Gate- and Bulk-Source-Voltage

Effective Channel Length and Width

Effective Channel Length

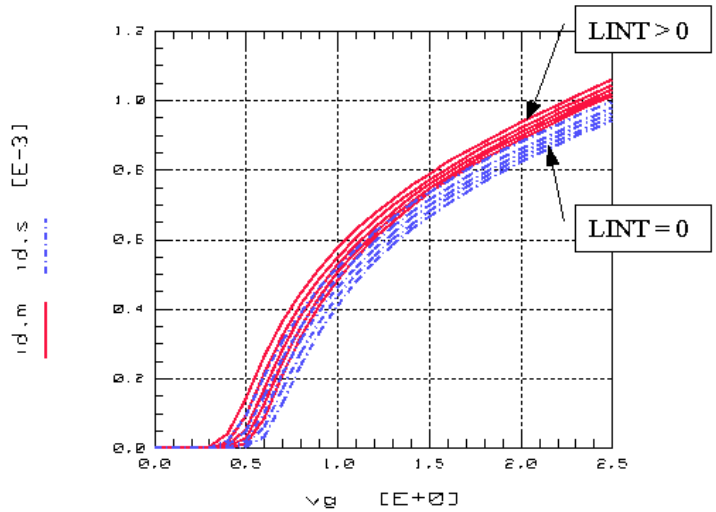


Figure 80 Influence of Channel Length Reduction on the Drain Current

The effective channel length is defined in BSIM3 as follows:

$$L_{eff} = L_{Designed} - 2dL \quad \textbf{(22)}$$

The channel length reduction on one side of the channel consists of several empirical terms as shown below:

$$dL = L_{int} + \frac{L_l}{L^{Lln}} + \frac{L_w}{W^{Lwn}} + \frac{L_{wl}}{L^{Lln} W^{Lwn}} \quad \textbf{(23)}$$

The use of the model parameters LL, LLN, LWN, LW and LWL is very critical because they are only used for fitting purposes. On the other hand, they may be needed to achieve a good fit over a large area of channel lengths especially for processes with a minimum designed gate length of less than 0.25µm. Figure 80 shows the influence of the geometrical

channel length reduction LINT on the drain current of a short channel transistor while Figure 81 represents the channel length reduction according to Equation 23.

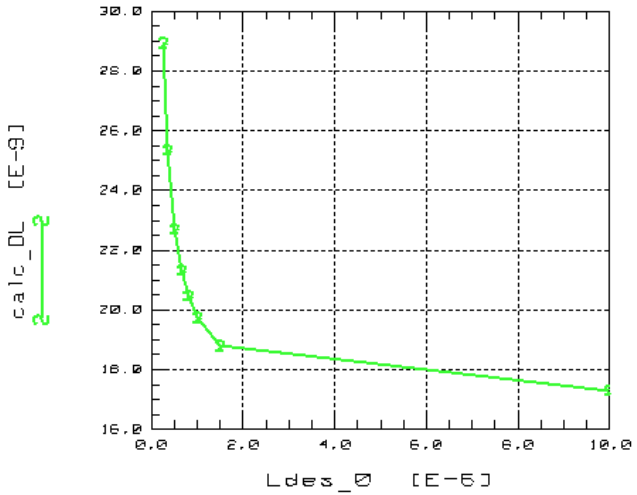


Figure 81 Channel Length Reduction dL as a Function of Channel Length L

Effective Channel Width

The effective channel width is defined in BSIM3 as follows:

$$W_{eff} = W_{Designed} - 2dW \tag{24}$$

The channel width reduction on one side of the channel consists of several empirical terms as shown below:

$$dW = W_{int} + \frac{W_l}{L^{Wln}} + \frac{W_w}{W^{Wwn}} + \frac{W_{wl}}{L^{Wln} W^{Wwn}} \tag{25}$$

The use of the model parameters WL, WLN, WWN, WW, and WWL is very critical because they are only used for fitting purposes. On the other hand, they may be needed to achieve a good fit over a large area of channel widths especially for

processes with a minimum designed gate width of less than 0.25µm. Figure 82 shows the influence of the geometrical channel width reduction WINT on the drain current of a narrow channel transistor while Figure 83 represents the channel width reduction according to Equation 25.

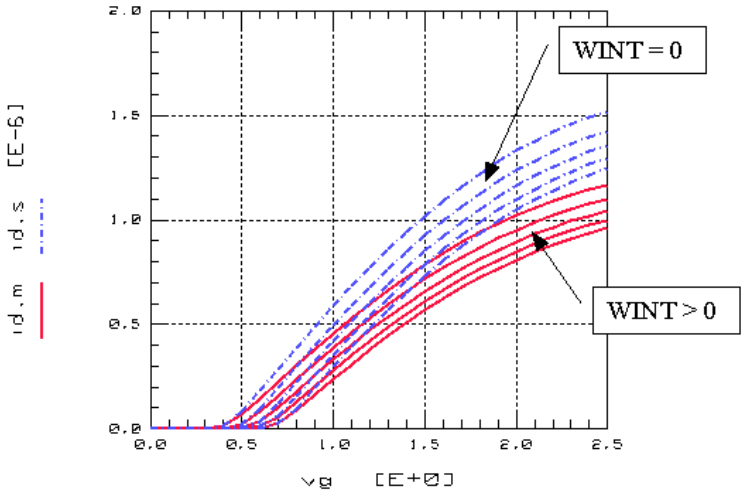


Figure 82 Influence of Channel Width Reduction on the Drain Current

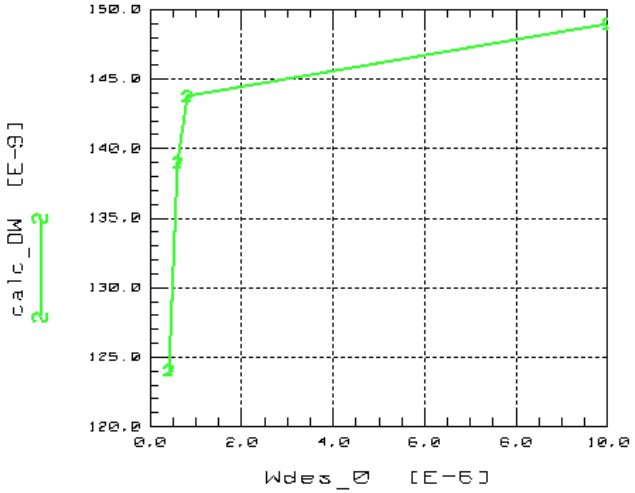


Figure 83 Channel Width Reduction dW as a Function of Channel Width W

Drain Current

Single Equation for Drain Current

In contrast to former implementations of the BSIM3 model, the drain current is represented through a single equation in all three areas of operation (subthreshold region, linear region, and saturation region). Due to this single formula, all first order derivatives of the drain current are continuous, which is an important prerequisite for analog simulations.

In the case that no parasitic drain/source resistance is given, the equation for the drain current is given below:

$$I_{ds0} = \mu_{eff} C_{ox} \frac{W}{L} \frac{V_{gsteff}\left(1 - A_{bulk}\frac{V_{dseff}}{2(V_{gsteff} + 2V_{tm})}\right)V_{dseff}}{1 + \frac{V_{dseff}}{E_{sat}L}} \quad \textbf{(26)}$$

This equation is valid for all three regions of operation of the MOS transistor because the voltages at drain, gate and bulk are replaced by effective drain voltage V_{dseff}, the effective gate voltage V_{gsteff} and the effective bulk voltage V_{bseff}, which are all defined by the continuous equations below:

Equation 27 shows the effective (V_{gs} - V_{th}) voltage, where the factor n is defined in Equation 31.

$$V_{gsteff} = \frac{2n\nu_t \, ln\left(1 + \exp\left(\frac{V_{gs} - V_{th}}{2n\nu_t}\right)\right)}{1 + 2nC_{ox}\sqrt{\frac{2\Phi_s}{q(\varepsilon_{si}N_{ch})}}\exp\left(-\frac{V_{gs} - V_{th} - 2V_{off}}{2n\nu_t}\right)} \quad \textbf{(27)}$$

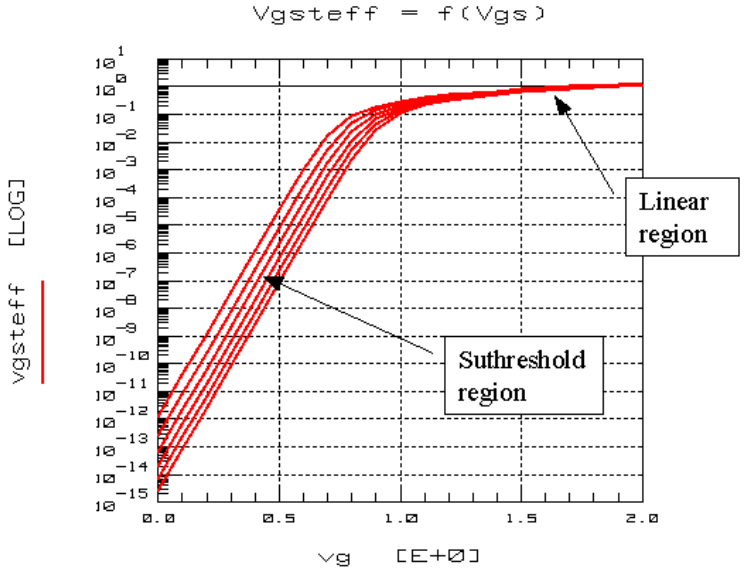


Figure 84 Effective Voltage V_{gs} - V_{th}

Figure 84 shows V_{gsteff} in logarithmic scale. V_{gsteff} fits a linear function for values of V_{gs} greater than V_{th} while the subthreshold area is covered by the fit of an exponential

function. Through this equation the first derivative is continuous between both operational regions (subthreshold and linear) of the MOS transistor.

Equation 28 shows the effective drain source voltage, V_{dseff}:

(28)

$$V_{dseff} = V_{dsat} - \frac{1}{2}\left(V_{dsat} - V_{ds} - \delta + \sqrt{(V_{dsat} - V_{ds} - \delta)^2 + 4\delta V_{dsat}}\right)$$

Figure 85 shows V_{dseff} in both the linear and the saturation region of operation of the MOS transistor. V_{dseff} models the transition between linear and saturation region without discontinuity in the first derivative of the drain current.

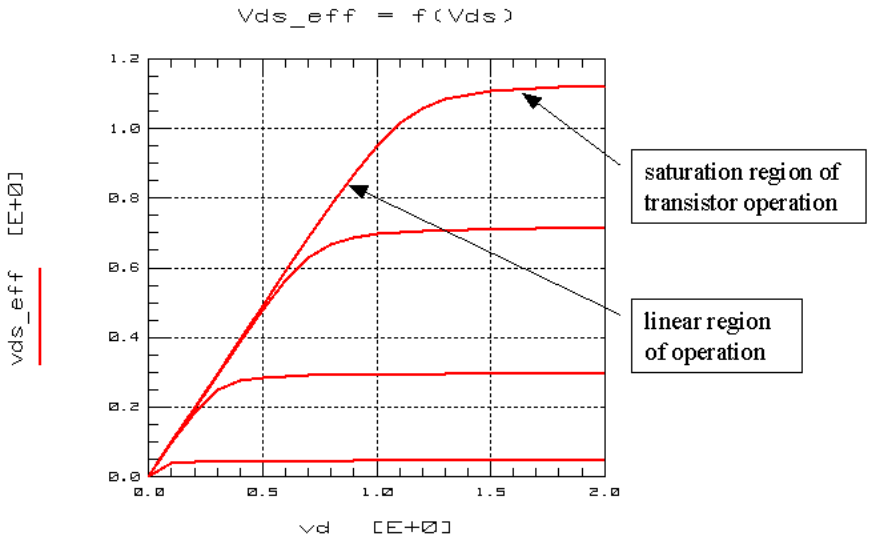


Figure 85 Effective Voltage V_{dseff}

Drain Saturation Voltage V_{dsat}

The equation for the drain saturation voltage is divided into two cases, the intrinsic case with $R_{ds} = 0$ and the extrinsic case with $R_{ds} > 0$:

$$V_{dsat} = \begin{cases} \dfrac{E_{sat} L_{eff} (V_{gsteff} + 2V_{tm})}{A_{bulk} E_{sat} L_{eff} + (V_{gsteff} + 2V_{tm})}, R_{ds} = 0 \\ \dfrac{-b - \sqrt{b^2 - 4ac}}{2a}, R_{ds} \neq 0 \end{cases} \quad \textbf{(29)}$$

where

$$a = A_{bulk}^2 R_{ds} C_{ox} W \nu_{sat} + \left(\frac{1}{\lambda} - 1\right) A_{bulk}$$

$$b = -(V_{gsteff} + 2V_{tm})\left(\frac{2}{\lambda} - 1\right) + A_{bulk} E_{sat} L_{eff} + 3A_{bulk} R_{ds} C_{ox} W \nu_{sat} (V_{gsteff} + 2V_{tm})$$

$$c = E_{sat} L_{eff} (V_{gsteff} + 2V_{tm}) + 2R_{ds} C_{ox} W \nu_{sat} (V_{gsteff} + 2V_{tm})^2$$

The influence of the maximum carrier velocity VSAT on the drain current I_{ds} and the conductance g_{ds} is demonstrated in Figure 86.

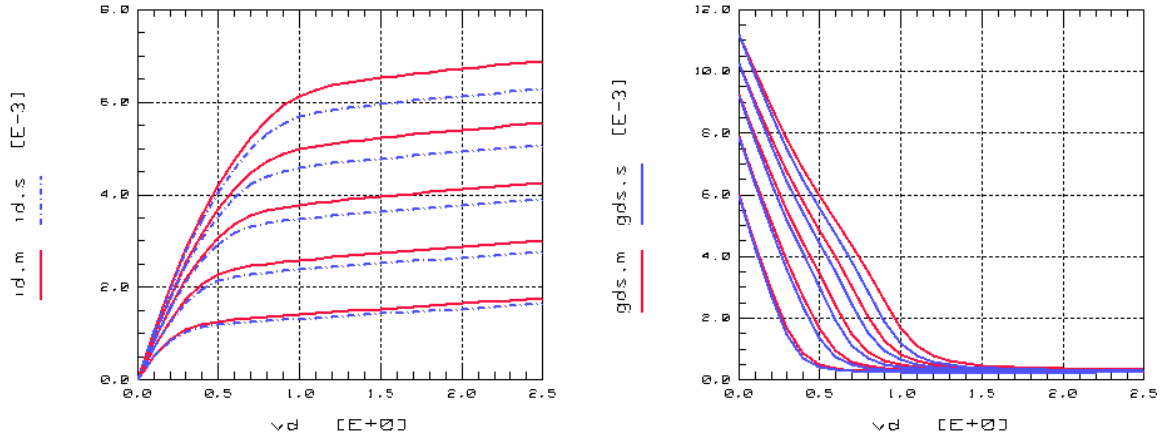


Figure 86 Influence of VSAT on Drain Current I_{ds} and Conductance g_{ds}

Bulk Charge Effect

When the drain voltage is high, combined with a long channel length, the depletion depth of the channel is not uniform along the channel length. This will cause the threshold voltage to vary along the channel length and is called bulk charge effect. Figure 87 shows the depletion depth as a function of channel length. For long channels, this effect causes a reduction of the drain current.

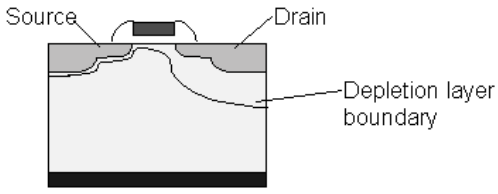


Figure 87 Depletion Width along the Channel Length

The bulk charge effect A_{bulk} is modeled in BSIM3 with the parameters A0, AGS, B0, B1, and KETA as shown in Equation 30.

(30)

$$A_{bulk} = \left(1 + \frac{K_1 \frac{T_{ox}}{T_{oxm}}}{2\sqrt{\Phi_s - V_{bseff}}}\left\{\frac{\frac{B_0}{W_{eff} + B_1} + \frac{A_0 L_{eff}}{L_{eff} + 2\sqrt{X_J X_{dep}}}}{\left[1 - AGS\, V_{gsteff}\left(\frac{L_{eff}}{L_{eff} + 2\sqrt{X_J X_{dep}}}\right)^2\right]}\right\}\right)\frac{1}{1 + K_{eta} V_{bs}}$$

The influence on the drain current is shown in Figure 88.

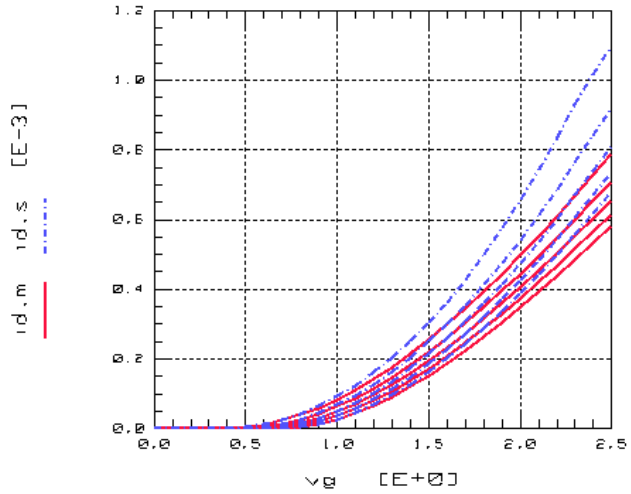


Figure 88 Influence of A0 and KETA on I_{ds} at High Drain Voltages

Drain Current in the Subthreshold Region

The drain current in the subthreshold region is modeled in BSIM3v3 by the effective voltage V_{gsteff}. The model parameters VOFF and NFACTOR describe the subthreshold current for a large transistor, while the parameters CDSC, CDSCD, and CDSCB are responsible for modeling the subthreshold behavior as a function of channel length. All these parameters contribute to the factor n in the formula for V_{gsteff} (see Equation 27).

$$n = 1 + N_{factor}\frac{C_d}{C_{ox}} + \frac{(C_{dsc} + C_{dscd}\,V_{ds} + C_{dscb}\,V_{bseff})}{C_{ox}}\theta_{th} + \frac{C_{it}}{C_{ox}} \tag{31}$$

$$\theta_{th} = e^{\left(-D_{VT1}\frac{L_{eff}}{2l_t}\right)} + 2e^{\left(-D_{VT1}\frac{L_{eff}}{l_t}\right)} \tag{32}$$

The influence of VOFF and NFACTOR on the drain current in the subthreshold region is shown in Figure 89.

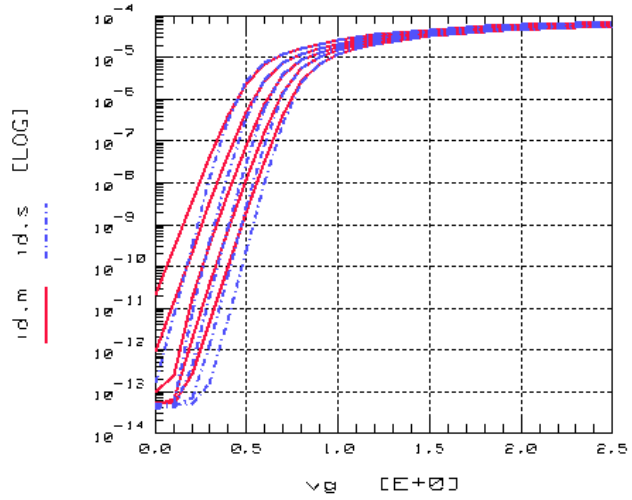


Figure 89 Influence of VOFF and NFACTOR on Drain Current in the Sub-threshold Region

Parasitic Resistance

As MOS devices are scaled into the deep submicron region, both the conductance g_m and the current of the device increase. Therefore the voltage drop across the source and drain series resistance becomes a non-negligible fraction of the applied drain source voltage. The resistance components associated with a MOSFET structure are shown in Figure 90. These include the contact resistance ($R_{contact}$) between metallization and source/drain area, the diffusion sheet resistance (R_{sheet}) of the drain/source area, the spreading resistance (R_{spread}) that arises from the current spreading from the channel, and the accumulation layer resistance ($R_{accum.}$).

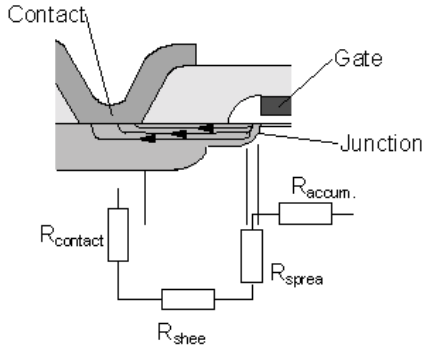


Figure 90 Resistance Components of a MOS Device

These components are put together to form the following equation in the BSIM3v3:

$$R_{ds} = \frac{R_{dsw}[1 + P_{rwg}V_{gsteff} + P_{rwb}(\sqrt{\Phi_s - V_{bseff}} - \sqrt{\Phi_s})]}{\left(10^6 W_{eff}\right)^{W_r}} \quad \textbf{(33)}$$

The diagram in Figure 91 visualizes the equation of R_{ds}. It should be noted that BSIM3 assumes that the drain resistance is equal to the source resistance. This symmetrical approach may cause difficulties if a device with a nonsymmetrical drain source resistance, for example a DMOS power transistor, should be modeled. In this case, a scalable SPICE macro model should add the required behavior to BSIM3.

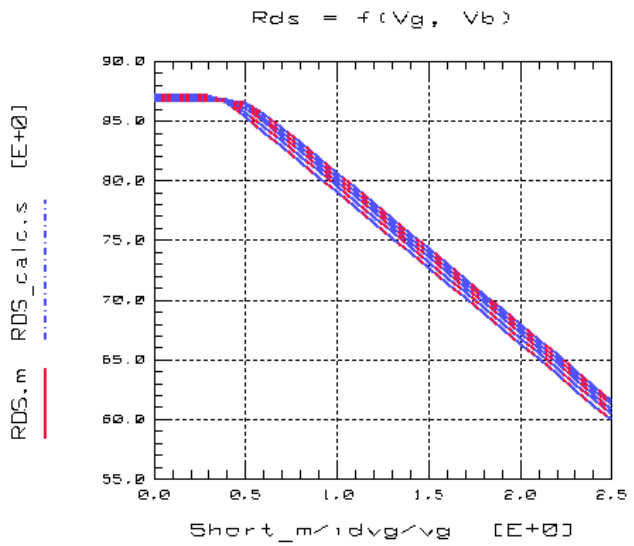


Figure 91 Drain Source Resistance R_{ds} as a Function of V_g and V_b

With this enhancement, Equation 26 for the drain current can be rewritten:

$$I_{ds} = \frac{I_{ds0}}{1 + R_{ds} I_{ds0} / V_{dseff}} \tag{34}$$

The influence of the parasitic resistance on the drain current is demonstrated for a SHORT and a SMALL transistor in Figure 92.

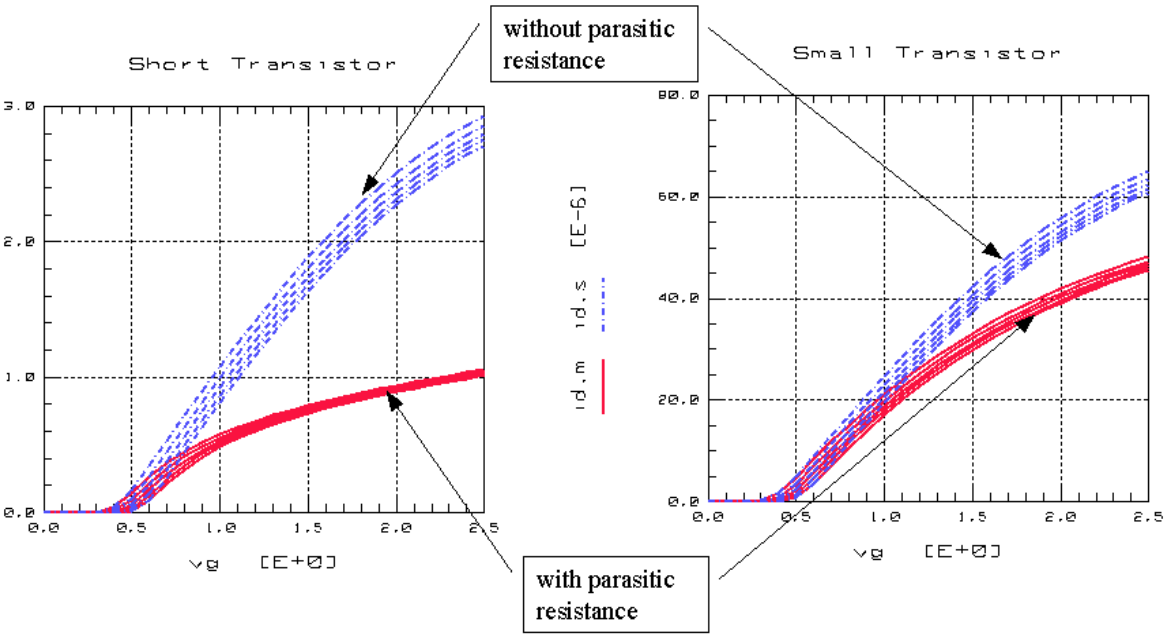


Figure 92 Influence of Drain Source Resistance on Drain Current

Output Resistance

a) Early Voltage

The drain current in the saturation region of submicron MOSFETs is influenced by the effects of channel length modulation (CLM), drain induced barrier lowering (DIBL), and substrate current induced body effect (SCBE). These effects can be seen clearly looking at the output resistance R_{out} of the device, which is defined as:

$$R_{out} = \frac{\delta V_{ds}}{\delta I_{ds}} \quad \textbf{(35)}$$

In Figure 93, the measured drain current and the output resistance of an n-type MOS transistor with a channel length of 0.5 µm are shown.

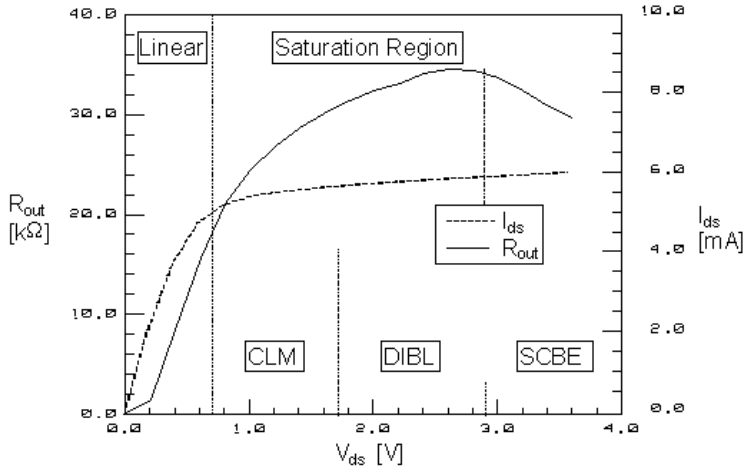


Figure 93 Drain Current and Output Resistance in Linear and Saturation Region

The left most region in Figure 93 is the linear region, in which carrier velocity is not saturated. The output resistance is small because the drain current has a strong dependence on the drain voltage. The other three regions belong to the saturation region. The three physical effects CLM, DIBL, and SCBE can be seen in the saturation region and are discussed in the following sections.

With the output resistance, the equation for the drain current (Equation 34) is enhanced by two additional terms and can be rewritten as:

(36)

$$I_{ds} = \frac{I_{ds0}}{1 + R_{ds} I_{ds0} / V_{dseff}} \left(1 + \frac{V_{ds} - V_{dseff}}{V_A}\right)\left(1 + \frac{V_{ds} - V_{dseff}}{V_{ASCBE}}\right)$$

The behavior of the output resistance is modeled in BSIM3 in the same way as the Early voltage of a bipolar transistor is modeled in the Gummel-Poon model. The Early voltage is divided in two parts, V_A due to DIBL and CLM and V_{ASCBE} due to SCBE. V_A is given by:

(37)

$$V_A = V_{Asat} + \left(1 + \frac{P_{vag} V_{gsteff}}{E_{sat} L_{eff}}\right)\left(\frac{1}{V_{ACLM}} + \frac{1}{V_{ADIBLC}}\right)^{-1}$$

where V_{Asat} is the Early voltage at V_{dsat}:

(38)

$$V_{Asat} = \frac{E_{sat} L_{eff} + V_{dsat} + 2R_{ds} \nu_{sat} C_{ox} W_{eff} V_{gsteff}\left(1 - \frac{A_{bulk} V_{dsat}}{2(V_{gsteff} + 2V_{tm})}\right)}{2/\lambda - 1 + A_{bulk} R_{ds} \nu_{sat} C_{ox} W_{eff}}$$

b) Channel length modulation (CLM)

When the drain bias approaches the drain saturation voltage, a region of high electric field forms near the drain and the electron velocity in this region saturates. In saturation, the length ΔL of the high-field region increases by an expansion in the direction of the source with increasing drain-source voltage V_{ds} and the MOSFET behaves as if the effective channel length has been reduced by ΔL. This phenomena is termed channel length modulation (CLM). CLM is not a special short-channel phenomenon, since the effect is present if a MOSFET is short or long. However, its relative importance increases and the effect on the saturated output conductance becomes distinctly more pronounced at shorter gate lengths.

The part of the Early voltage due to CLM is given by:

$$V_{ACLM} = \frac{1}{PCLM} \frac{A_{bulk} E_{sat} L + V_{gsteff}}{A_{bulk} E_{sat} l} (V_{ds} - V_{dseff}) \quad \textbf{(39)}$$

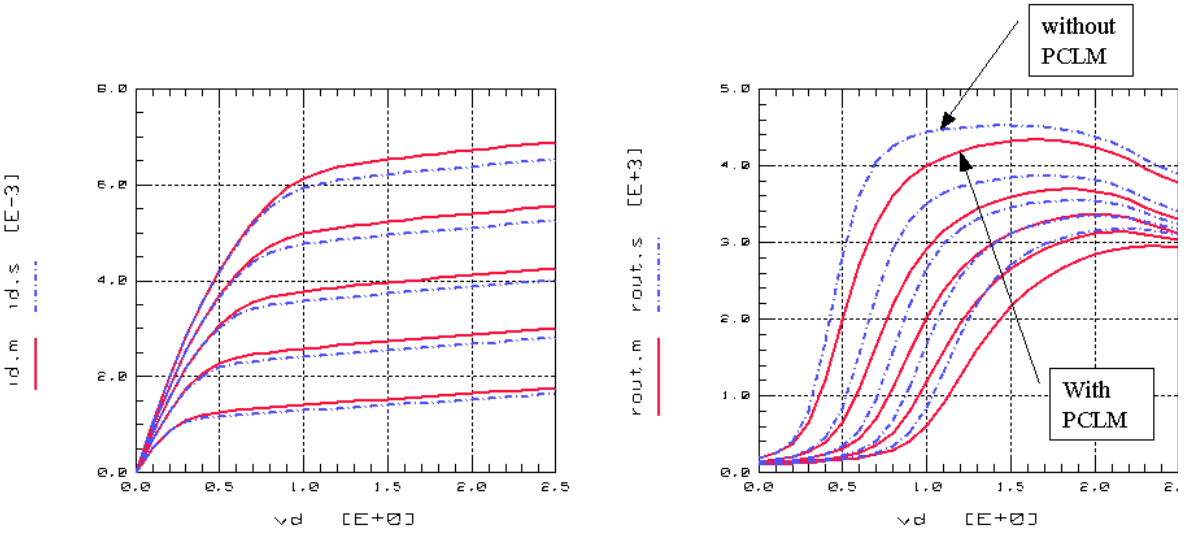


Figure 94 Channel Length Modulation (CLM)

c) Drain Induced Barrier Lowering (DIBL)

The depletion charges near source and drain are under the shared control of these contacts and the gate. In a short-channel device, this shared charge will constitute a relatively large fraction of the total gate depletion charge and can be shown to give rise to an increasingly large shift in the threshold voltage V_{th} with decreasing channel length L. Also, the shared depletion charge near drain expands with increasing drain-source bias, resulting in an additional V_{ds} dependent shift in V_{th}. This effect is related to a drain voltage induced lowering of the injection barrier between the source and the channel and is termed the drain induced barrier lowering (DIBL). Figure 95 shows the band diagram at the semiconductor-insulator interface of an 0.1 µm n-channel MOSFET simulated by a device simulator. The symmetrical profiles correspond to V_{ds} = 0 and the asymmetrical profiles to V_{ds} > 0. In the figure, the simulated

potential barrier near the source is observed to decrease with increasing drain bias, which indicates the origin of the DIBL effect.

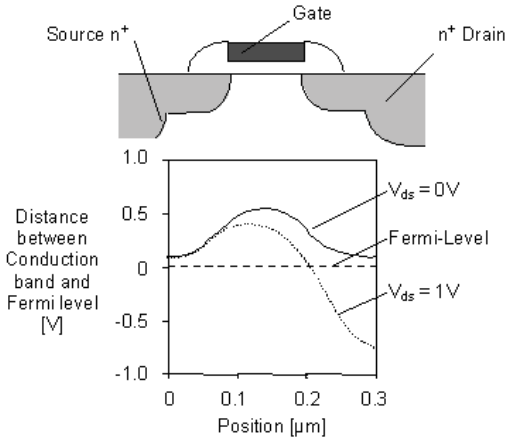


Figure 95 Band Diagram at Si-SiO_2 Interface of a 0.1 µm MOSFET

The DIBL effect is modeled in BSIM3v3 with the following equations:

(40)

$$V_{ADIBLC} = \frac{(V_{gsteff} + 2V_{tm})}{\Theta_{rout}(1 + P_{DIBLC}V_{bseff})}\left[1 - \left(\frac{A_{bulk}V_{dsat}}{A_{bulk}V_{dsat} + V_{gsteff} + 2V_{tm}}\right)\right]$$

with

(41)

$$\Theta_{rout}(L) = P_{DIBLC1}\left[\exp\left(-\frac{D_{rout}L_{eff}}{2l_{t0}}\right) + 2\exp\left(-\frac{D_{rout}L_{eff}}{l_{t0}}\right)\right] + P_{DIBLC2}$$

Figure 96 shows the influence of the DIBL effect on the output resistance of a short channel transistor.

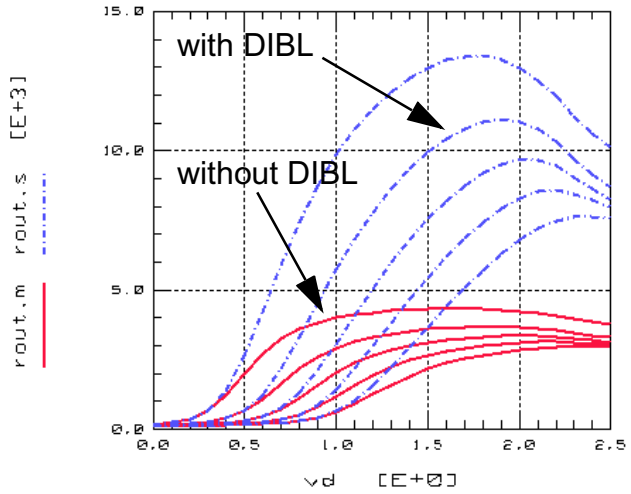


Figure 96 Influence of Drain Induced Barrier Lowering (DIBL) effect on output resistance

d) Substrate Current Induced Body Effect (SCBE)

Substrate current is induced through hot electrons at high drain voltages, as described in "Substrate Current" on page 230. It is suggested that the substrate current increases exponentially with the applied drain voltage. The total drain current will change, because it is the sum of the channel current from the source as well as the substrate current. It can be expressed as:

$$I_{ds} = I_{source} + I_{bulk} \quad \textbf{(42)}$$

The increase of the total drain current through hot electrons will be described by the part V_{ASCBE} of the Early voltage which results in a lowering of the output resistance for high drain voltage (Figure 97).

$$V_{ASCBE} = \left[\frac{P_{SCBE2}}{L}\exp\left(-\frac{P_{SCBE1}\,l}{(V_{ds}-V_{dsat})}\right)\right]^{-1} \quad \textbf{(43)}$$

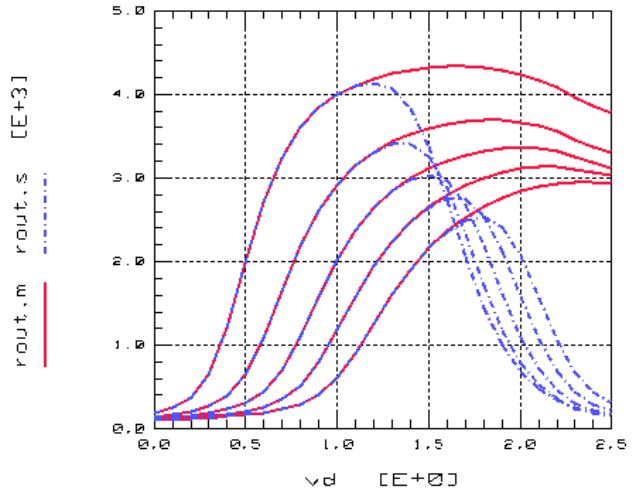


Figure 97 Substrate Current Body Effect (SCBE)

Substrate Current

In a n-channel MOSFET, electrons in the channel experience a very large field near the drain. In this high field, some electrons coming from the source will be energetic enough to cause impact ionization, and additional electrons and holes are generated by avalanche multiplication. The high energy electrons are referred as *hot* electrons. The generated electrons are attracted to the drain, adding to the channel current, while holes are collected by the substrate contact, resulting in a substrate current, which is shown in Figure 98.

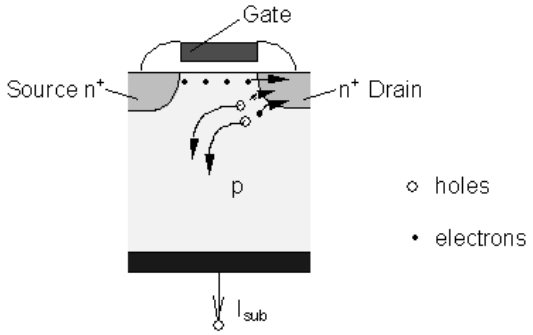


Figure 98 Generation of Substrate Current in an n-channel MOSFET

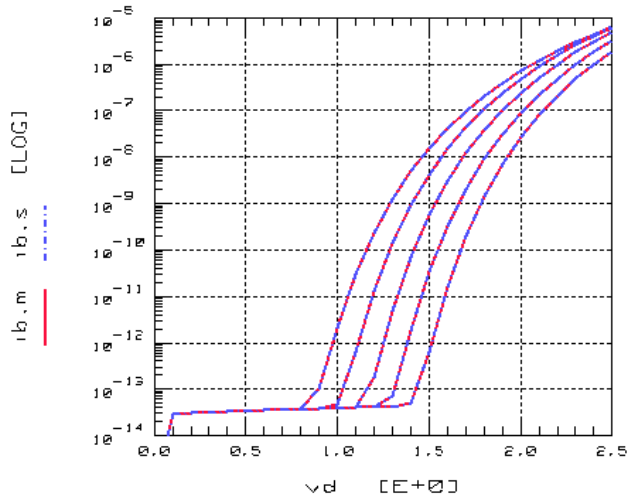


Figure 99 Substrate Current I_{bs} parameterized by V_g

The substrate current is described in BSIM3 by the following equation:

$$I_{sub} = \frac{\alpha_0 + \alpha_1 L_{eff}}{L_{eff}}(V_{ds} - V_{dseff}) \exp\left(-\frac{\beta_0}{V_{ds} - V_{dseff}}\right) I_{ds}\left(1 + \frac{V_{ds} - V_{dseff}}{V_A}\right) \quad \textbf{(44)}$$

Drain / Bulk and Source / Bulk Diodes

Figure 100 shows a pn-junction diode between the bulk and the drain of an n-type MOS Transistor.

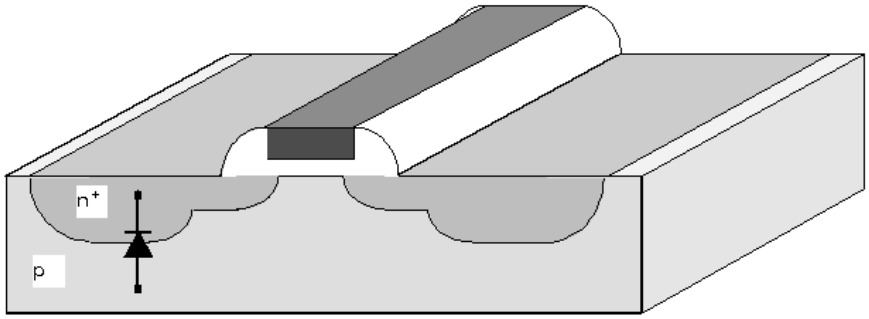


Figure 100 pn-junction diode

The drain/bulk and the source/bulk pn-junctions can be used as diodes in CMOS designs. BSIM3v3 offers a simple DC model for the current I_{bs} or IBD flowing through these diodes.

$$I_{bs} = \begin{cases} I_{sbs}\left(e^{\left(\frac{V_{bs}}{NV_{tm}}\right)} - 1\right) + G_{MIN}V_{bs} \\ IJTH + \frac{IJTH + I_{sbs}}{NV_{tm}}(V_{bs} - V_{jsm}) + G_{MIN}V_{bs} \end{cases} \quad \textbf{(45)}$$

where NJ is the emission coefficient of the source junction and the saturation current I_{sbs} is calculated as:

$$I_{sbs} = A_S J_S + P_S \quad \textbf{(46)}$$

where J_S is the saturation current density of the source/bulk diode, A_S is the area of the source junction, J_{SSW} is the sidewall saturation current density of the source/bulk diode, and P_S is the perimeter of the source junction. J_S and J_{SSW} are functions of the temperature and can be described as:

$$J_S = J_{S0} e^{\left(\frac{\frac{E_{g0}}{V_{tm0}} - \frac{E_g}{V_{tm}} + X_{TI} \ln\left(\frac{T}{T_{nom}}\right)}{NJ} \right)} \quad \textbf{(47)}$$

$$J_{SSW} = J_{S0SW} e^{\left(\frac{\frac{E_{g0}}{V_{tm0}} - \frac{E_g}{V_{tm}} + X_{TI} \ln\left(\frac{T}{T_{nom}}\right)}{NJ} \right)} \quad \textbf{(48)}$$

where:

$$E_{g0} = 1.16 - \frac{7.02 \times 10^{-4} Tnom^2}{Tnom + 1108}$$

$$E_g = 1.16 - \frac{7.02 \times 10^{-4} T^2}{T + 1108}$$

J_{S0} is the saturation current density (default is 10^{-4} A/m^2)

J_{S0SW} is the sidewall saturation current density (default is 0)

$NV_{tm} = NJ \cdot (K_b T/q)$

$V_{jsm} = NV_{tm} \ln (ijth/I_{sbs} + 1)$

The current I_{bs} through the diode is shown in the following figure:

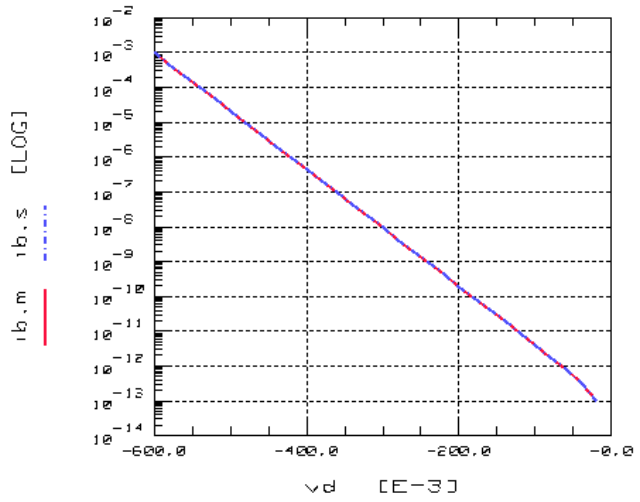


Figure 101 Current I_{bs} Through Diode

Consistency Check of DC measurement data for multiple measured devices

You can perform a quick consistency check of the measured data versus gate length, gate width, and temperature. If there are measurement errors, they can be easily identified using this additional check of DC measurement data.

Drain Saturation Current Idsat

Displaying the absolute values of ID_{SAT} versus the gate length of all measured devices does not easily show measurement errors because the absolute currents spread all over the diagram, as shown in the left part of Figure 102.

In this diagram, absolute values of ID_{SAT} versus L and W are displayed. ID_{SAT} is determined at max. Vg, max. Vd, and Vb=0 for one temperature. Each number represents one transistor and each color a different value of the transistors gate width W. For example, transistors marked 4 have a gate width of 5 microns and 9 different L values, as shown by the number of 4's in the following figure.

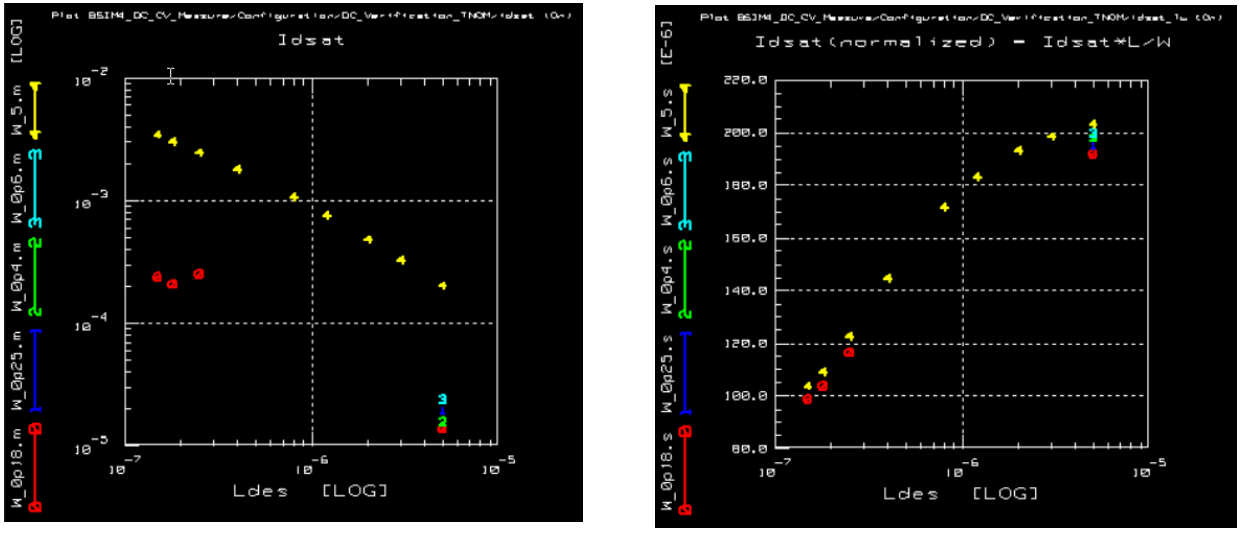
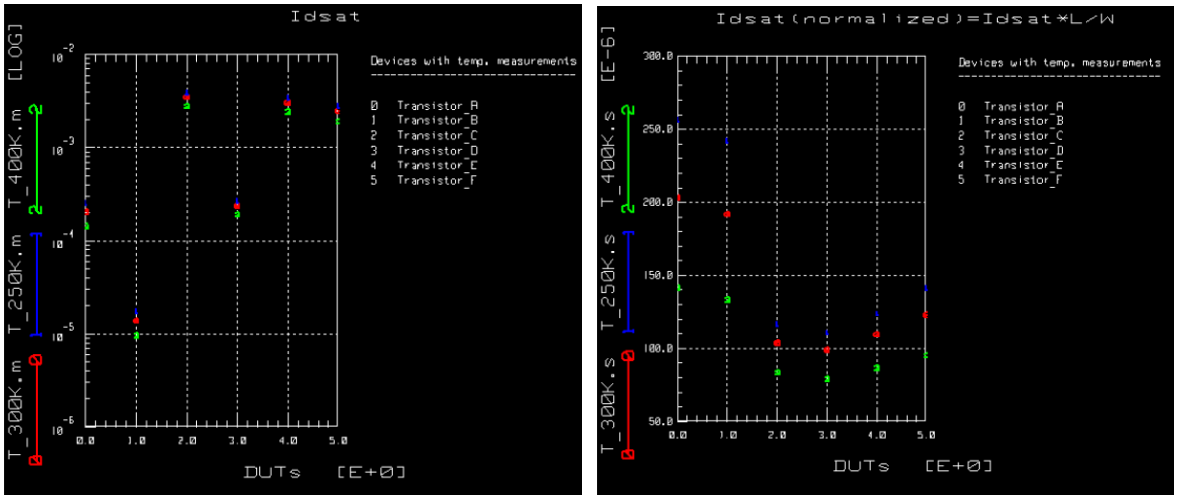


Figure 102 Left part: ID_{SAT} = f(W, L); right part: $ID_{SATnorm}$ = f(W, L)

But if the same values (measured at the same temperature) are displayed in a normalized representation $ID_{SATnorm}$ = Idsat*L/W (see right part of Figure 102), the values appear in a sorted way. They are shown from the transistors having the highest gate width values on top of the lower gate width transistors. The transistors having the smallest gate width values are shown at the lowest display position in the diagram.

If the temperature measurements of the transistors are normalized as well, the measured data is again sorted. The following diagram shows ID_{SAT} and $ID_{SATnorm}$ for devices with temperature measurements. Each color represents one temperature and each value of the x-axis represents one device. To allow a correlation to the device name, the plot annotation feature has been used to display a corresponding list of device names.



Left part: ID_{SAT} = f(temp, device); right part: $ID_{SATnorm}$ = f(temp, device)

Threshold voltage

Similar normalized data representations are available for the threshold voltage Vth of measured devices, see Figure 103. Vth is determined for each device at Vb=0 and low Vd. The following diagram shows Vth as a function of L, W (left part), and temperature (right part) for those devices. Vth is determined using the reference current method:

$$V_{th} = V_G(I_{D0})$$

with: $I_{D0} = I_{Dref} \cdot \frac{W}{L}$, using $I_{Dref} = 100nA$

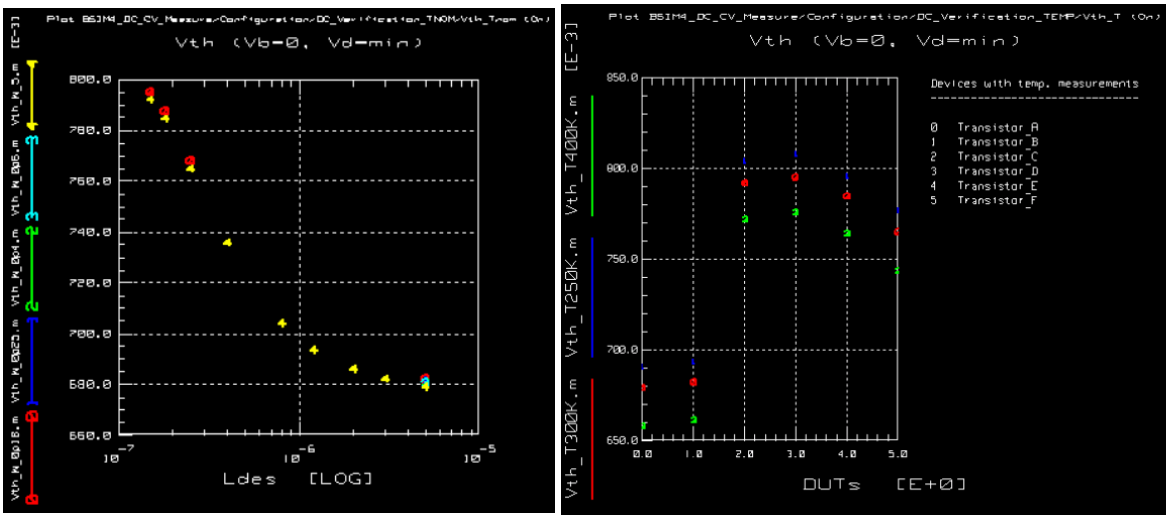


Figure 103 Left part: V_{th} = f(L, W); Right part: V_{th} = f(temp, device)

Capacitance Model

Please use the model *bsim3_tutor_cv.mdl* provided with the BSIM3v3 Modeling Package to visualize the capacitance model parameters. Load the file into IC-CAP and run the different macros to see how certain parameters affect the device behavior of a deep submicron MOS transistor.

The capacitance in a MOS transistor can be divided into three different parts:

- Junction capacitance C_{Junc} between source/drain and the bulk region
- Capacitance of the extrinsic MOS transistor which consists of:
  - The outer fringing capacitance C_F between polysilicon gate and the source/drain
  - The overlap capacitance C_{GDO} between the gate and the heavily doped source/drain regions
  - The overlap capacitance C_{GDOL} between the gate and the lightly doped source/drain regions
- Capacitance of the intrinsic MOS transistor in the region between the metallurgical source and drain junction when the gate is at flat band voltage.

These different parts of the capacitance of a MOS transistors are shown in Figure 104. The following three sections explain each type of capacitance and its implementation in the BSIM3v3 model.

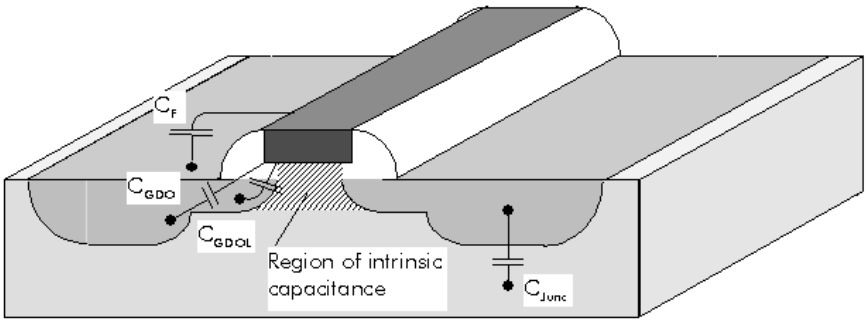


Figure 104 Different Parts of the Capacitance of a MOS Transistor

Junction Capacitance

The source/drain-bulk junction capacitance can be divided into three components as shown in Figure 105. The calculation is shown for the drain-bulk junction capacitance. The source-bulk capacitance is calculated in the same way with the same model parameters.

The overall junction capacitance C_{jdb} is given by:

$$C_{jdbs} = \begin{cases} C_{AREA} + C_{SW} + C_{SWG} & \text{if PS} > Weff \\ C_{AREA} + C_{SW} & \text{if PS} < Weff \end{cases} \quad \textbf{(49)}$$

where:

C_{AREA} is the bottom area capacitance

C_{SW} is the sidewall or peripheral capacitance along the three sides of the junction's field oxide

C_{SWG} is the sidewall or peripheral capacitance along the gate oxide side of the junction

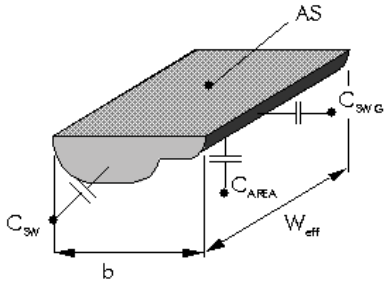


Figure 105 Dimensions of Drain/Source Region and Different Capacitance Parts

Bottom area capacitance C_{AREA}

$$C_{AREA} = AD^{*}C_{jbd} \quad \textbf{(50)}$$

where:

AD area of bottom side of pn junction, given as SPICE model parameter

C_{jbd} capacitance per unit area of the drain-bulk junction

C_{jbd} is calculated according to the following equation and is shown in Figure 106.

For $V_{bs} < 0$:

$$C_{jbs} = C_j\left(1 - \frac{V_{bs}}{P_b}\right)^{-M_j} \quad \textbf{(51)}$$

For $V_{bs} \geq 0$:

$$C_{jbs} = C_j\left(1 + M_j\frac{V_{bs}}{P_b}\right)$$

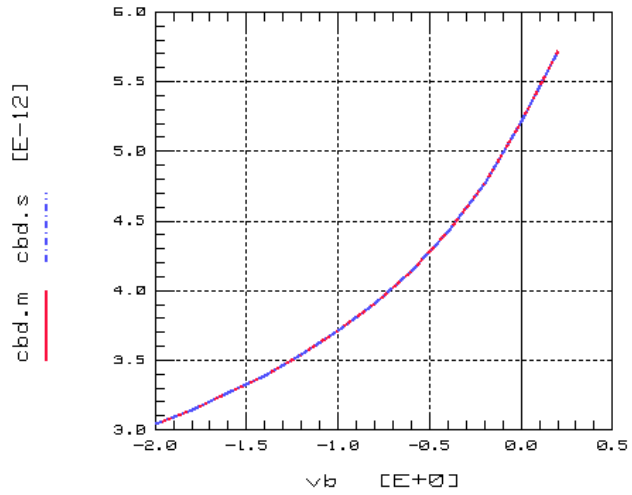


Figure 106 Bottom Area Capacitance C_{jbd} as a Function of V_g

Peripheral sidewall capacitance C_{SW} along the field oxide

$$C_{SW} = (PD - W_{eff})C_{jbdsw} \quad \textbf{(52)}$$

where:

PD	total perimeter of pn junction, given as SPICE model parameter
W_{eff}	effective gate width of transistor, calculated in SPICE
C_{jbdsw}	capacitance per unit length

C_{jbdsw} is calculated according to the following equation and is shown in Figure 107:

For $V_{bs} < 0$:

$$C_{jbdsw} = C_{jsw}\left(1 - \frac{V_{bs}}{P_{bsw}}\right)^{-M_{jsw}} \quad \textbf{(53)}$$

For $V_{bs} \geq 0$:

$$C_{jbdsw} = C_{jsw}\left(1 + M_{jsw}\frac{V_{bs}}{P_{bsw}}\right)$$

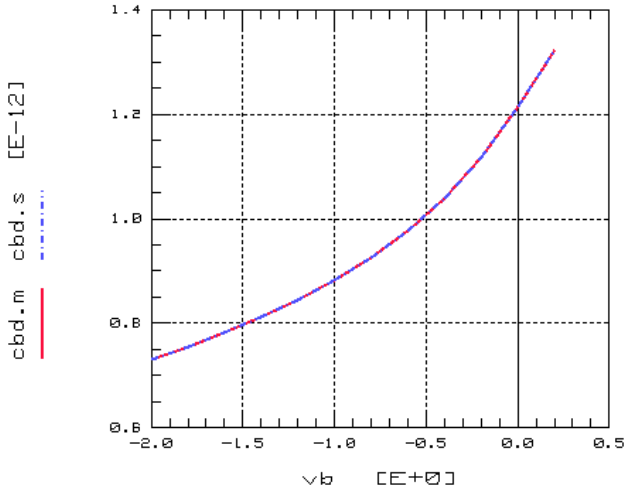


Figure 107 Sidewall Capacitance C_{jbdsw} as a Function of V_g

Peripheral sidewall capacitance C_{SWG} along the gate oxide

$$C_{SWG} = W_{eff}C_{jbdswg} \quad \textbf{(54)}$$

where:

W_{eff}	effective gate width of transistor, calculated in SPICE
C_{jbdswg}	capacitance per unit length

C_{jbdswg} is calculated according to the following equation and is shown in Figure 108.

For $V_{bs} < 0$:

$$C_{jbs} = C_{jswg}\left(1 - \frac{V_{bs}}{P_{bswg}}\right)^{-M_{jswg}} \quad \textbf{(55)}$$

For $V_{bs} \geq 0$:

$$C_{jbs} = C_{jswg}\left(1 + M_{jswg}\frac{V_{bs}}{P_{bswg}}\right)$$

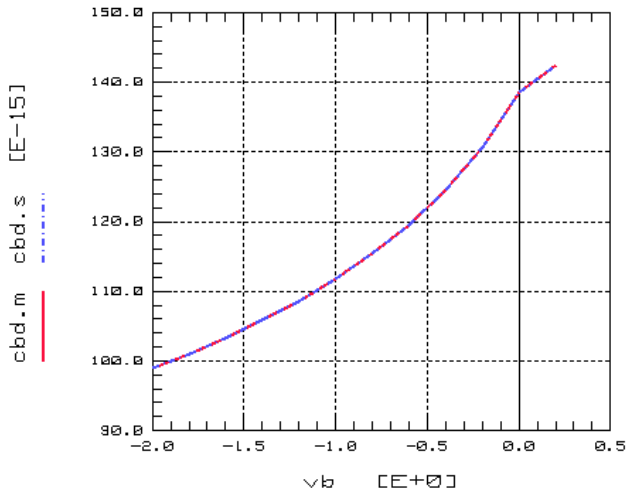


Figure 108 Sidewall Capacitance C_{jbdswg} Along the Gate Oxide as a Function of V_g

Extrinsic Capacitance

As mentioned in the introduction to this chapter, the extrinsic capacitance of a MOS transistor consists of the following three components:

- the outer fringing capacitance C_F between polysilicon gate and the source/drain
- the overlap capacitance C_{GDO} between the gate and the heavily doped source/drain regions
- the overlap capacitance C_{GDOL} between the gate and the lightly doped source/drain regions

The contribution of these different components to the overall extrinsic capacitance is demonstrated in Figure 109 and Figure 110.

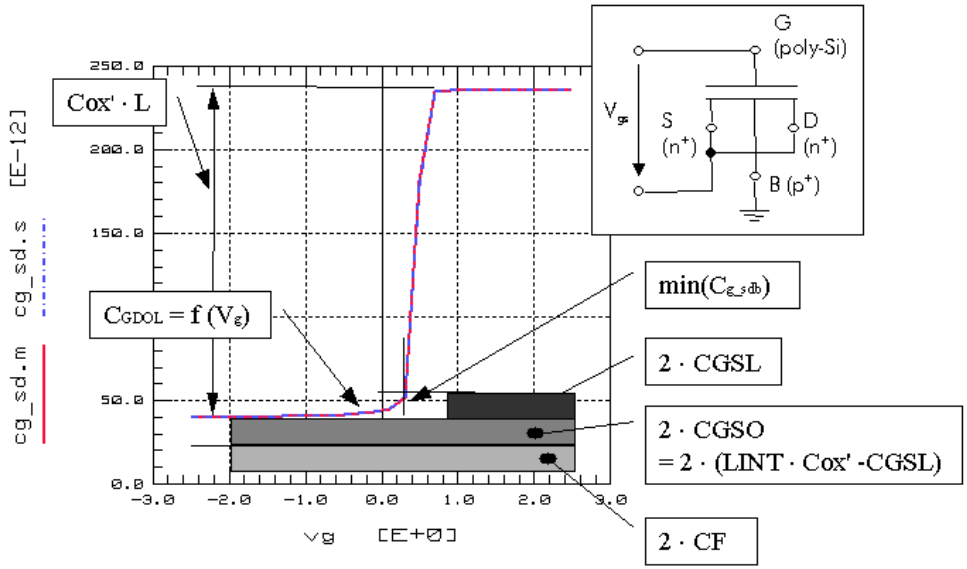


Figure 109 Different Components of the Extrinsic Capacitance

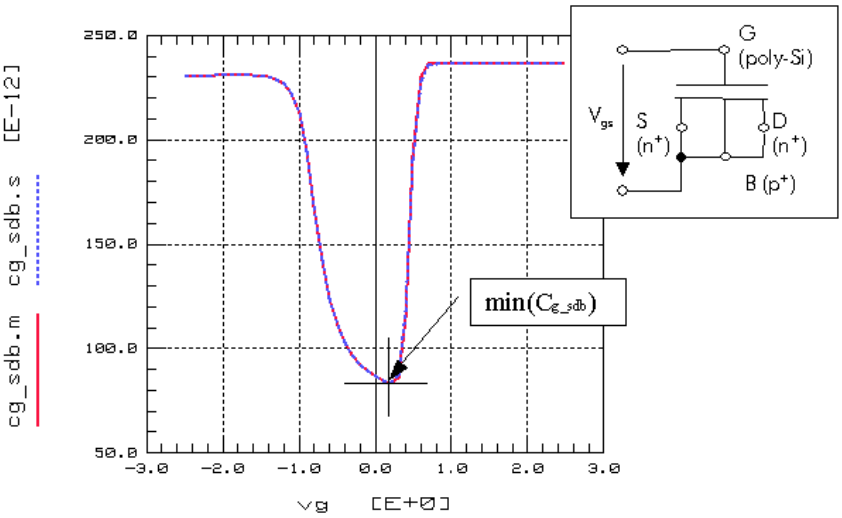


Figure 110 Overlap Capacitance Between Gate and Drain/Source/Bulk

a) Fringing Capacitance

The fringing capacitance of a MOS transistor consists of a bias independent outer fringing capacitance and a bias dependent inner fringing capacitance. In the present release of the BSIM3v3 model, only the bias independent outer fringing capacitance is implemented. Experimentally, it is virtually impossible to separate this capacitance with the overlap capacitance. Nonetheless if the model parameter CF is not given, the outer fringing capacitance can be calculated with the following equation:

$$CF = \frac{2\varepsilon_{SiO_2}}{\pi} ln\left(1 + \frac{4 \times 10^{-7}}{T_{ox}}\right) \quad \textbf{(56)}$$

b) Overlap Capacitance

In BSIM3v3 an accurate model for the overlap capacitance is implemented. In old capacitance models this capacitance is assumed to be bias independent. However, experimental data show that the overlap capacitance changes with gate to

source and gate to drain biases. In a single drain structure or the heavily doped S/D to gate overlap region in a LDD structure, the bias dependence is the result of depleting the surface of the source and drain regions. Since the modulation is expected to be very small, this region can be modeled with a constant capacitance. However in LDD MOSFETs a substantial portion of the LDD region can be depleted, both in the vertical and lateral directions. This can lead to a large reduction of overlap capacitance. This LDD region can be in accumulation or depletion.

In BSIM3v3, a single equation is implemented for both regions by using such smoothing parameters as $Vgs_{overlap}$ and $Vgd_{overlap}$ for the source and drain side, respectively. Unlike the case with the intrinsic capacitance, the overlap capacitances are reciprocal. In other words, $Cgs_{overlap} = Csg_{overlap}$ and $Cgd_{overlap} = Cdg_{overlap}$.

The model equations for the overlap capacitance are shown for the drain overlap capacitance and are identical for the source overlap capacitance:

Overlap charge per gate width:

(57)

$$\frac{Q_{overlap}}{W_{eff}} = CGDOV_{gs} + CGDL\left\{V_{gd} - V_{gd,\,overlap} - \frac{CKAPPA}{2}\left(-1 + \sqrt{1 - \frac{4V_{gd,\,overlap}}{CKAPPA}}\right)\right\}$$

where:

$$CKAPPA = \frac{2\varepsilon_{si} qN_{LDO}}{C_{ox}^2}$$

with the smoothing parameter:

$$V_{gd,\,overlap} = \frac{1}{2}\left((V_{gd} + \delta_2) - \sqrt{(V_{gd} + \delta_2)^2 + 4\delta_2}\right) \quad \textbf{(58)}$$

$$\delta_2 = 2/100$$

for the measurement and simulation conditions given in Figure 109, this results in the overlap capacitance:

$$C_{gd,\,overlap} = \frac{\partial Q_{overlap}}{\partial V_{gs}} \tag{59}$$

The model parameter CGDO in Equation 57 can be calculated by the following equation:

$$CGDO = (DLC \cdot C_{OX}) - CGDL \tag{60}$$

where DLC represents the channel length reduction in the BSIM3v3 capacitance model. Please see the next section for more details about DLC:

Intrinsic Capacitance

a) Geometry for Capacitance Model

The BSIM3v3 model uses different expressions for the effective channel length L_{eff} and the effective channel width W_{eff} for the I-V and the C-V parts of the model.

The geometry dependence for the intrinsic capacitance part is given as the following:

$$\Delta W = DWC + \frac{WL}{L^{WLN}} + \frac{WW}{W^{WWN}} + \frac{WWL}{L^{WLN} W^{WWN}} \tag{61}$$

$$\Delta L = DLC + \frac{LL}{L^{LLN}} + \frac{LW}{W^{LWN}} + \frac{LWL}{L^{LLN} W^{LWN}} \tag{62}$$

L_{active} and W_{active} are the effective length and width of the intrinsic device for capacitance calculations. The parameter ΔL is equal to the source/drain to gate overlap length plus the difference between drawn and actual poly gate length due to processing (gate printing, etching, and oxidation) on

one side. The L_{active} parameter extracted from the capacitance method is a close representation of the metallurgical junction length (physical length).

$$W_{active} = W_{Drawn} - 2\Delta W \quad \textbf{(63)}$$

$$L_{active} = L_{Drawn} - 2\Delta L \quad \textbf{(64)}$$

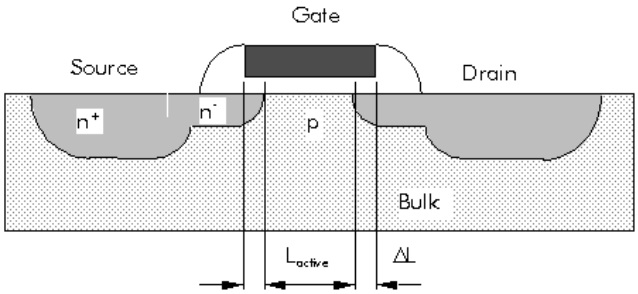


Figure 111 Dimensions of a MOSFET

While the authors of the BSIM3v3 model suggest to use a parameter LINT for the I-V model, which is different from DLC, other literature sources [3] propose that LINT should have the same value as DLC. This approach is also implemented in the BSIM3v3 Modeling Package to ensure that the extracted values of the channel length reduction are very close to the real device physics. Therefore, the channel length reduction LINT for the I-V model will be set to DLC from the C-V model extracted from capacitance measurements.

b) Intrinsic Capacitance Model

The intrinsic capacitance model that is implemented in the BSIM3 model is based on the principle of conservation of charge. There are a few major considerations in modeling the intrinsic capacitance of a deep submicron MOS transistor:

- The difficulty in capacitance measurement, especially in the deep submicron regime. At very short channel lengths, the MOSFET intrinsic capacitance is very small while the conductance is large.
- Charge can only be measured at high impedance nodes (i.e., the gate and substrate nodes), only 8 of the 16 capacitance components in an intrinsic MOSFET can be directly measured. An alternative solution is to use a 2-D device simulator.
- The access to the internal charges in a simulator.

Therefore, this section presents no details about the intrinsic charge formulations. Please refer to the BSIM3v3 manual [1] for more information. Only the basic principles are described here.

To ensure charge conservation, terminal charges instead of the terminal voltages are used as state variables. The terminal charges Q_g, Q_b, Q_s, and Q_d are the charges associated with the gate, bulk, source, and drain. The gate charge is comprised of mirror charges from 3 components:

- The channel minority (inversion) charge (Q_{inv})
- The channel majority (accumulation) charge (Q_{acc})
- The substrate fixed charge (Q_{sub})

The accumulation charge and the substrate charge are associated with the substrate node while the channel charge comes from the source and drain nodes:

$$\begin{aligned} Q_g &= -(Q_{sub} + Q_{inv} + Q_{acc}) \\ Q_b &= Q_{sub} + Q_{acc} \\ Q_{inv} &= Q_s + Q_d \end{aligned} \quad \textbf{(65)}$$

The inversion charges are supplied from the source and drain electrodes. The ratio of Q_d and Q_s is the charge partitioning ratio. Existing charge partitioning schemes are 0/100, 50/50 and 40/60 (given by the model parameter XPART = 0, 0.5, and 1) which are the ratios of Q_d to Q_s in the saturation region.

From these four terminal charges, 9 transcapacitances $C_{(terminal,voltage)}$ are calculated inside the BSIM3 model as partial derivatives with respect to the voltages V_{gb}, $V_{db,}$ and V_{sb}. The abbreviation can be interpreted as:

C_{ggb} ..... partial derivative of Q_g with respect to V_{gb}

Partial derivatives of Q_g:

$$C_{ggb} = \frac{\partial Q_g}{\partial V_{gb}}$$
$$C_{gdb} = \frac{\partial Q_g}{\partial V_{db}} \quad \textbf{(66)}$$
$$C_{gsb} = \frac{\partial Q_g}{\partial V_{sb}}$$

Partial derivatives of Q_d:

$$C_{dgb} = \frac{\partial Q_d}{\partial V_{gb}}$$
$$C_{ddb} = \frac{\partial Q_d}{\partial V_{db}} \quad \textbf{(67)}$$
$$C_{dsb} = \frac{\partial Q_d}{\partial V_{sb}}$$

Partial derivatives of Q_b:

$$C_{bgb} = \frac{\partial Q_b}{\partial V_{gb}}$$
$$C_{bdb} = \frac{\partial Q_b}{\partial V_{db}} \quad \textbf{(68)}$$
$$C_{bsb} = \frac{\partial Q_b}{\partial V_{sb}}$$

The 9 transcapacitances previously introduced are shown in the following three plots for a simulation setup as shown in the following figure:

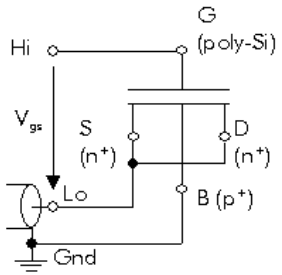


Figure 112 Simulation and Measurement Setup for Overlap Capacitances

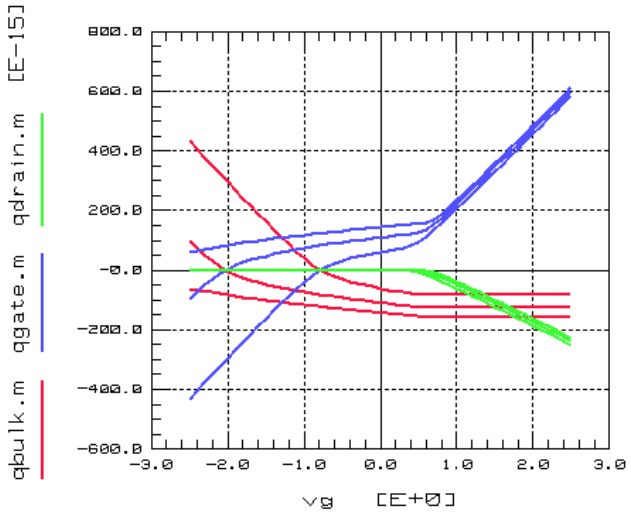


Figure 113 Terminal charges Q_g, Q_b and Q_d

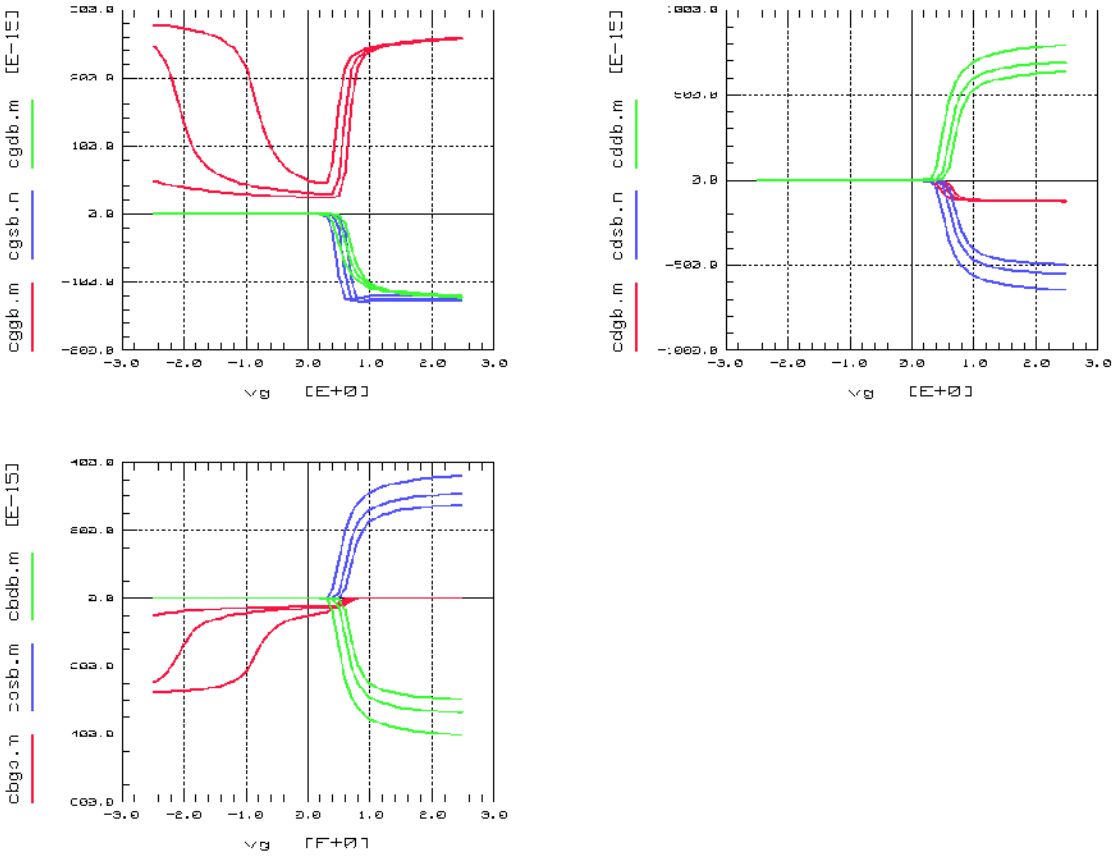


Figure 114 Partial derivatives of Q_g, Q_b and Q_d with respect to V_{db}, V_{gb} and V_{sb}

The Overall Capacitance in BSIM3

In previous sections, the three components of the BSIM3 capacitance model were introduced. Now when an AC simulation is performed the capacitance, which can be *measured* at the terminals, is composed of different parts of junction capacitances, extrinsic capacitances, and intrinsic capacitances.

The following figure shows, as an example, the capacitance components for the overlap capacitance between gate and bulk/source/drain as simulated according to the following circuit description:

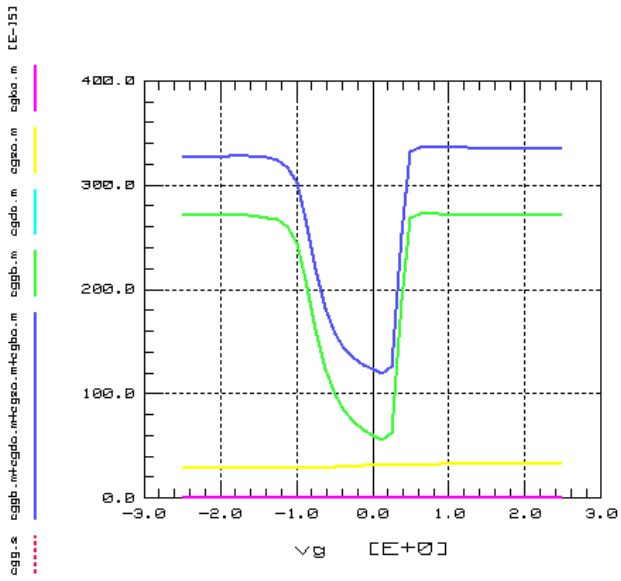


Figure 115 Different Parts of Overlap Capacitance C_Gate_SDB

The overlap capacitance C_Gate_SDB consists of:

$$C_{jbs} = C_j\left(1 + M_j\frac{V_{bs}}{P_b}\right) \quad \textbf{(69)}$$

where:

C_{ggb}	intrinsic capacitance
$C_{gd,overlap}$	overlap capacitance between gate and drain
$C_{gs,overlap}$	overlap capacitance between gate and source
$C_{gb,overlap}$	overlap capacitance between gate and bulk

Other capacitances can be calculated in the same way. Please refer to the BSIM3 manual for more details.

High Frequency Behavior

Macro Model for High Frequency Application

Using the BSIM3v3 model for the simulation of high frequency applications requires a major change in the model structure. A new concept of a SPICE simulation model for deep submicron devices based on the standard BSIM3v3.2.4 model was found, which is able to satisfy a correct DC simulation and the representation of the RF behavior of the MOS devices. Figure 116 shows the subcircuit used for RF simulation using the BSIM3 model together with an explanation of the physical structure responsible for each element of the subcircuit.

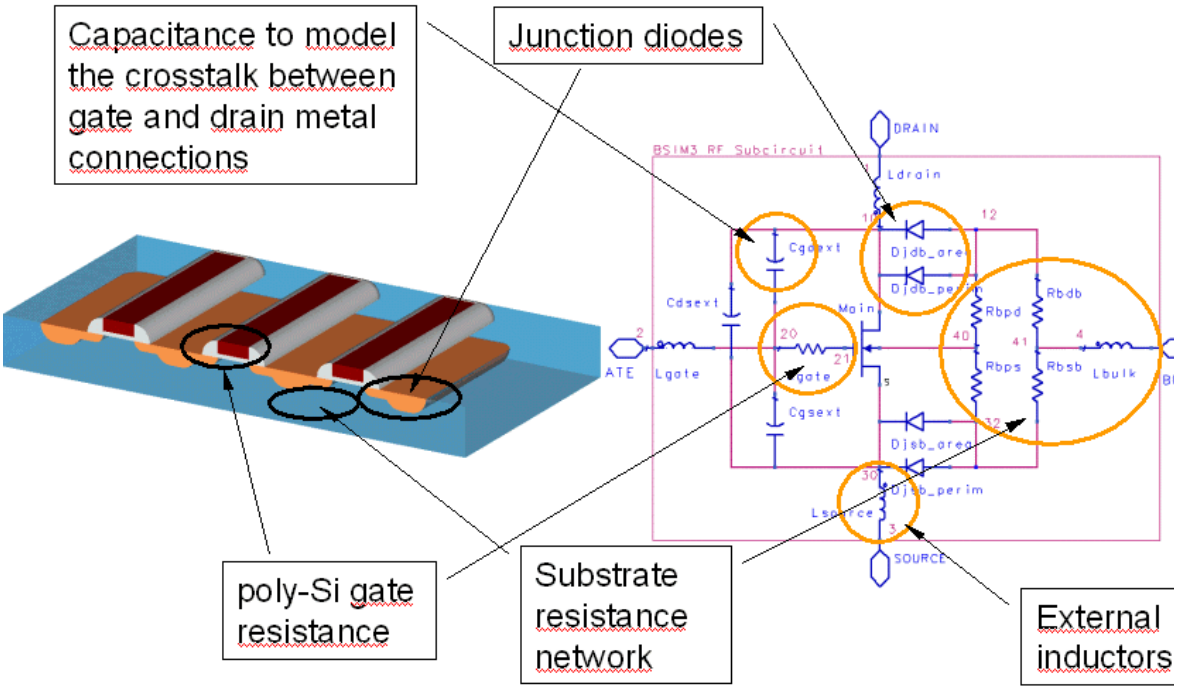


Figure 116 Equivalent Circuit for the SPICE Macro Model

The model itself is implemented as the macro model shown above—no changes are done in the BSIM3v3.2.4 model code itself. This is the ultimate precondition for its use in a commercial circuit simulator that includes the BSIM3v3.2.4 model and makes it available to circuit design engineers. The BSIM3v3.2.4 model already consists of a non-quasi-static model and an accurate capacitance model, which makes it the ideal base for RF simulations. However, the description of the resistance behavior of a transistor is very poor. In the BSIM3v3.2.4 model itself, no gate resistance is included. Due to the nature of the MOS transistor, such a resistance cannot be seen in the DC operation region. However, looking at the real existing poly silicon gates of modern MOS devices, there is a resistance which cannot be neglected in AC simulations. This resistance, R_{gate}, has a major influence on the reflection coefficient S11 of an input signal to the MOS transistor as demonstrated in Figure 117.

It should be noted that the parameter R_{gate} in this high frequency model is used to fit the input reflection of the MOS transistor. Therefore, it is very likely that R_{gate} has a different value as the measured sheet resistance of the poly-Si gate during process characterization on PCMs using for instance a van-der-Pauw test structure.

The second enhancement in the RF BSIM3v3.2.4 macro model is a resistance network for the substrate resistance, which is described by four resistors R_{BPD}, R_{BPS}, R_{BDB}, and R_{BSB} [7, 8]. The substrate resistance can be seen in the reverse-reflection coefficient S22 at the output of the transistor. Together with the resistance network, the internal drain-bulk and source-bulk junction diodes of the BSIM3v3.2.4 model are replaced by the external elements Djdb_area, Djdb_perim, Djsb_area, and Djsb_perim. The decoupling diodes account for the same voltage dependant values of the bottom and the sidewall capacity as the internal junction capacitances. This replacement is the prerequisite for a correct modeling of the substrate resistance.

With this approach, the model is valid for both the DC and the RF behavior of the transistor.

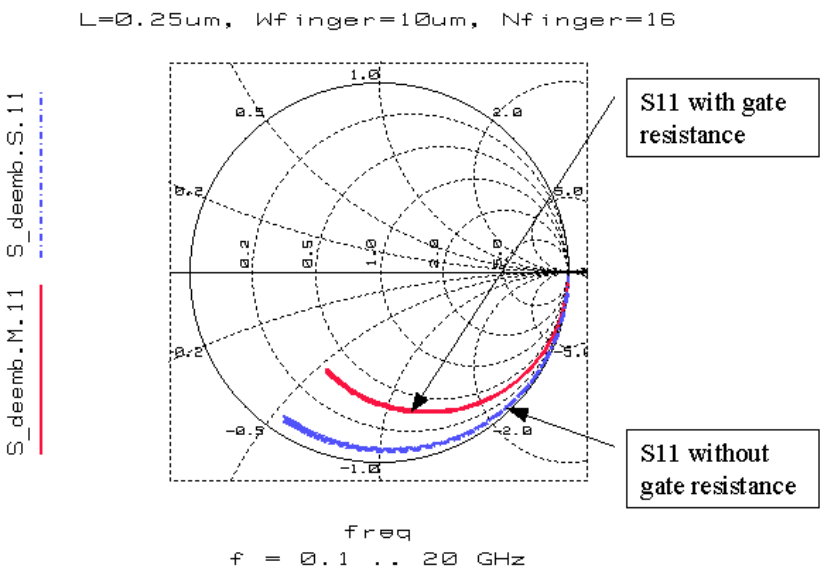


Figure 117 Influence of gate resistance on input reflection S11

Single Subcircuit Model for BSIM3v3 RF Transistors

The macro model approach results in a subcircuit for single RF MOS transistors, which the following circuit file shows (according to Figure 116).

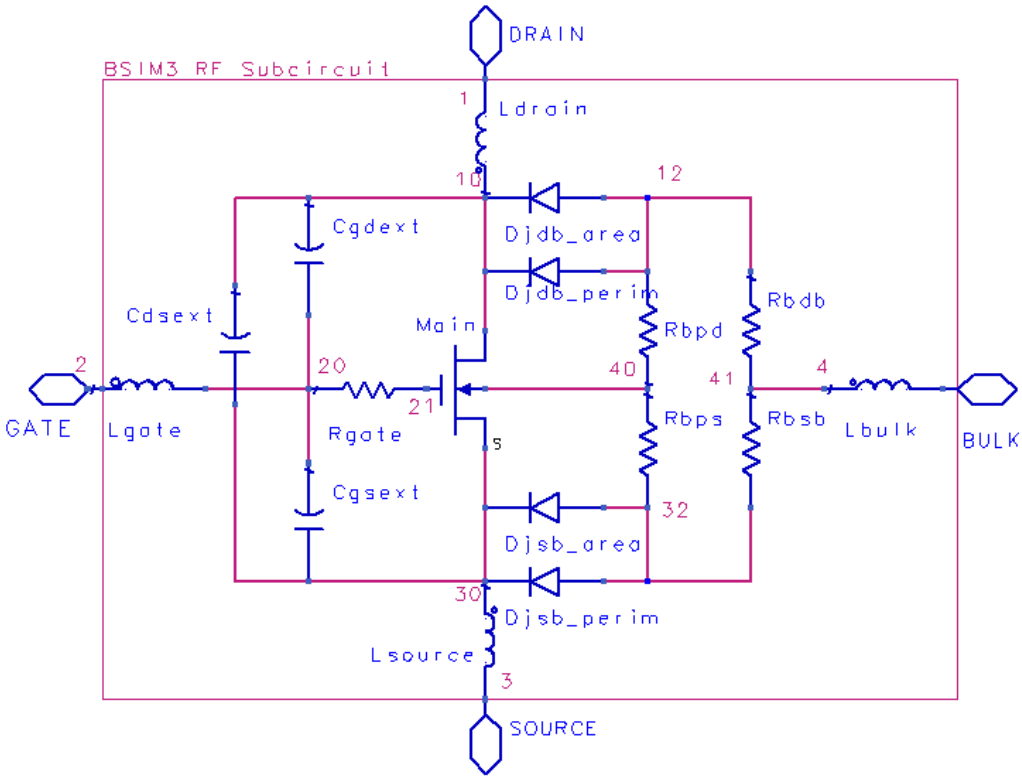


Figure 118 Subcircuit for RF modeling of single transistors using the BSIM3-model

Following is part of the SPICE netlist used for single transistors in BSIM3v3 RF modeling.

```
LINK CIRC Circuit
{
data
{
circuitdeck
{
 .OPTIONS GMIN=1.0E-14
 * ------------------------------------------------------------------------
 * Single subcircuit model for BSIM3v3.2.4 RF n-type devices
 *
 * Simulator: UCB Spice3e2
 * Model: BSIM3 Modeling Package
 * Date: 25.04.2003
 * Origin: ICCAP_ROOT/..../bsim3/code/circuits/spice3/cir/rf_nmos_single.cir
 * ------------------------------------------------------------------------
 *
 .subckt bsim3_rf_extract 1 2 3 4
 *
 *--- BSIM3 model card ------------------------------------------------------
#echo .MODEL BSIM3_HF NMOS
#echo + LEVEL=$mpar(LEVEL=8) VERSION=3.2.4 BINUNIT=$mpar(BINUNIT=2)
#echo + MOBMOD=$mpar(MOBMOD=1 CAPMOD=$mpar(CAPMOD=3) NOIMOD=$mpar(NOIMOD=1)
#echo + PARAMCHK=$mpar(PARAMCHK=1)
#echo + DELTA=$mpar(DELTA=0.01)TNOM=$mpar(TNOM=27)TOX=$mpar(TOX=7.5E-9)
#echo + TOXM=$mpar(TOXM=7.5E-9)
#echo + NCH =$mpar(NCH=1.7e17) XJ=$mpar(XJ=1.5E-7) NGATE=$mpar(NGATE=0) RSH =$mpar(RSH=0)
#echo + VTH0=$mpar(VTH0=0.7) K1=$mpar(K1=0.53) K2=$mpar(K2=-0.013) K3=$mpar(K3=0)
#echo + K3B =$mpar(K3B=0) W0=$mpar(W0=2.5E-6) NLX =$mpar(NLX=0.174u) DVT0=$mpar(DVT0=2.2)
#echo + DVT1=$mpar(DVT1=0.53) DVT2=$mpar(DVT2=-0.032) DVT0W=$mpar(DVT0W=0)
#echo + DVT1W=$mpar(DVT1W=5.3E6)
#echo + DVT2W=$mpar(DVT2W=-0.032) ETA0=$mpar(ETA0=0) ETAB=$mpar(ETAB=0)
#echo + DSUB=$mpar(DSUB=0.56)
#echo + U0=$mpar(U0=670) UA=$mpar(UA=2.25E-9) UB=$mpar(UB=5.87E-19) UC=$mpar(UC=4.65E-11)
#echo + VSAT=$mpar(VSAT=8e4) A0=$mpar(A0=1) AGS =$mpar(AGS=0) B0=$mpar(B0=0)
#echo + B1=$mpar(B1=0) KETA=$mpar(KETA=-0.047) A1=$mpar(A1=0) A2=$mpar(A2=1)
#echo + RDSW=$mpar(RDSW=0) PRWB=$mpar(PRWB=0) PRWG=$mpar(PRWG=0) WR=$mpar(WR=1)
#echo + WINT=$mpar(WINT=0) WL=$mpar(WL=0) WLN =$mpar(WLN=1) WW=$mpar(WW=0)
#echo + WWN =$mpar(WWN=1) WWL =$mpar(WWL=0) DWG =$mpar(DWG=0) DWB =$mpar(DWB=0)
#echo + LINT=$mpar(LINT=0) LL=$mpar(LL=0) LLN =$mpar(LLN=1) LW=$mpar(LW=0)
#echo + LWN =$mpar(LWN=1) LWL =$mpar(LWL=0)
#echo + VOFF=$mpar(VOFF=-0.08) NFACTOR=$mpar(NFACTOR=1) CIT =$mpar(CIT=0)
#echo + CDSC=$mpar(CDSC=2.4E-4)
#echo + CDSCB=$mpar(CDSCB=0) CDSCD=$mpar(CDSCD=0) PCLM=$mpar(PCLM=1.3)
#echo + PDIBLC1=$mpar(PDIBLC1=0.39)
#echo + PDIBLC2=$mpar(PDIBLC2=0.0086) PDIBLCB=$mpar(PDIBLCB=0.0) DROUT=$mpar(DROUT=0.56)
#echo + PSCBE1=$mpar(PSCBE1=4.24E8)
#echo + PSCBE2=$mpar(PSCBE2=1.0E-5) PVAG=$mpar(PVAG=0) VBM =$mpar(VBM=-3)
#echo + ALPHA0=$mpar(ALPHA0=0) ALPHA1=$mpar(ALPHA1=0) BETA0=$mpar(BETA0=30)
#echo + JS=1e-20 JSW =1.0E-20 NJ=1 IJTH=$mpar(IJTH=0.1)
#echo + CJ=0 MJ=0.5 PB=1 CJSW=0
#echo + MJSW=0.33 PBSW=1 CJSWG=$mpar(CJSWG=5E-10) MJSWG=$mpar(MJSWG=0.33)
#echo + PBSWG=$mpar(PBSWG=1) CGDO=$mpar(CGDO=0) CGSO=$mpar(CGSO=0)
#echo +CGBO=$mpar(CGBO=0)
```

```
#echo +CGBO=$mpar(CGBO=0) CGSL=$mpar(CGSL=0) CGDL=$mpar(CGDL=0)
#echo + CKAPPA=$mpar(CKAPPA=0.6) CF=$mpar(CF=0)
#echo + NOFF=$mpar(NOFF=1) VOFFCV=$mpar(VOFFCV=0) ACDE=$mpar(ACDE=1)
#echo + MOIN=$mpar(MOIN=15) DLC=$mpar(DLC=0) DWC=$mpar(DWC=0)
#echo + LLC=$mpar(LLC=0) LWC=$mpar(LWC=0) LWLC=$mpar(LWLC=0)
#echo + WLC=$mpar(WLC=0) WWC=$mpar(WWC=0) WWLC=$mpar(WWLC=0)
#echo + CLC=$mpar(CLC=0.1E-6) CLE=$mpar(CLE=0.6) ELM=$mpar(ELM=2)
#echo + XPART=$mpar(XPART=0.5) KT1=$mpar(KT1=-0.11) KT1L=$mpar(KT1L=0)
#echo + KT2=$mpar(KT2=0.022) UTE=$mpar(UTE=-1.5) UA1=$mpar(UA1=4.31E-9)
#echo + UB1=$mpar(UB1=-7.6E-18) UC1=$mpar(UC1=-5.6E-11) AT=$mpar(AT=3.3E4)
#echo + PRT=$mpar(PRT=0) XTI=$mpar(XTI=3.0) TPB=$mpar(TPB=0)
#echo + TPBSW=$mpar(TPBSW=0) TPBSWG=$mpar(TPBSWG=0)
#echo + TCJ=$mpar(TCJ=0) TCJSW=$mpar(TCJSW=0) TCJSWG=$mpar(TCJSWG=0)
#echo + AF=$mpar(AF=1.5) EF=$mpar(EF=1.5) KF=$mpar(KF=1e-17) EM=$mpar(EM=4.1E7)
#echo + NOIA=$mpar(NOIA=2e29) NOIB=$mpar(NOIB=5e4) NOIC=$mpar(NOIC=-1.4e-12)
*--- Parasitic diode model cards ------------------------------------------
#echo .MODEL bsim_diode_area D
#echo + CJO=$mpar(CJ=5E-4) VJ=$mpar(PB=1) M=$mpar(MJ=0.5)
#echo + IS=$mpar(JS=1.0E-4) N=$mpar(NJ=1)
#echo .MODEL bsim_diode_perim D CJO=$mpar(CJSW=5E-10) VJ=$mpar(PBSW=1)
#echo + M=$mpar(MJSW=0.33) IS=$mpar(JSW=1.0E-12) N=$dpar(CALC.NJSW=1)
* --------- Gate network ---------------------------------------------------
CGDEXT 20 10 0.1f
CGSEXT 20 30 0.1f
RGATE 20 21 100
LGATE 2 20 1p
* --------- Drain network --------------------------------------------------
CDSEXT 10 30 0.1f
LDRAIN 1 10 1p
* --------- Source network -------------------------------------------------
LSOURCE 3 30 1p
* --------- Substrate network ----------------------------------------------
* Diodes are for n-type MOS transistors
#echo Djdb_area 12 10 bsim_diode_area AREA=$dpar(x_rf_transistor.AD=10e-12)
#echo Djdb_perim 12 10 bsim_diode_perim AREA=$dpar(x_rf_transistor.PD=22e-6)
#echo Djsb_area 32 30 bsim_diode_area AREA=$dpar(x_rf_transistor.AS=10e-12)
#echo Djsb_perim 32 30 bsim_diode_perim AREA=$dpar(x_rf_transistor.PS=22e-6)
RBDB 12 40 100
RBSB 32 40 100
RBPD 12 41 100
RBPS 32 41 100
LBULK 4 40 1p
*--- call single MOSFET ---------------------------------------------------
#echo MAIN 10 21 30 41 BSIM3_HF
#echo + L=$dpar(x_rf_transistor.L=1u) W=$dpar(x_rf_transistor.W=10e-6)
#echo + AD=$dpar(x_rf_transistor.AD=10e-12) AS=$dpar(x_rf_transistor.AS=10e-12)
#echo + PD=$dpar(x_rf_transistor.PD=22e-6) PS=$dpar(x_rf_transistor.PS=22e-6)
#echo + NRS=$dpar(x_rf_transistor.NRS=0) NRD=$dpar(x_rf_transistor.NRD=0)
#echo + NQSMOD=$mpar(NQSMOD=0)
.ends
}
}
}
```

The single BSIM3 RF model represents exactly one measured test device, the substrate resistance network uses fixed values for the resistors R_{BPS}, R_{BPD}, R_{BDB}, and R_{BSB}. Also, the external parasitics (L, C, ...) are fixed and valid only for one measured device. Using this approach, the parameters extracted are valid only for a specific transistor geometry, which means you must have RF parameters for each of your possible device geometries. This requires measurement and library creation for every transistor geometry in your design.

Design engineers often need to have scalable transistors for easy design processes. Therefore, extensions are made to use a scalable BSIM3 model. Those are described in the following section.

Fully Scalable Subcircuit Model for BSIM3v3 RF Transistors

Figure 119 shows a cross section of a multifinger RF MOSFET with the distances marked from the bulk connection point to the physical transistor connections.

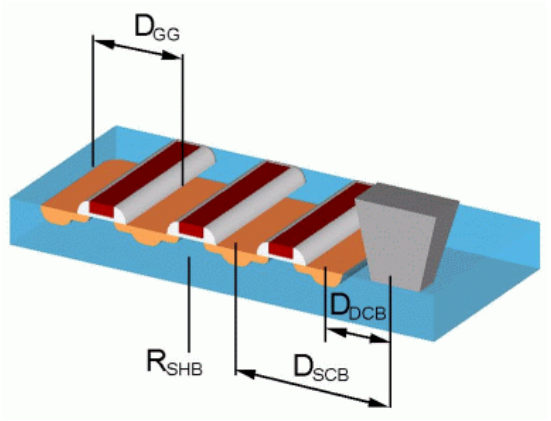


Figure 119 Distances between the bulk connection and the terminals of a multifinger RF transistor

The distances are:

- D_{DCB}: distance between bulk connection point and drain

- D_{DCS}: distance between bulk connection point and source
- D_{GG}: distance between gate stripes

Additionally, the sheet resistance of the bulk connection, R_{SHB}, is needed.

Implementations according to “MOS Transistor Modeling for RF IC Design” [8] and our own findings to model a scalable substrate resistance behavior are leading to the following equations, which are implemented into the SPICE subcircuit for the fully scalable BSIM3 RF model.

Using the distances according to Figure 119, the resistors are calculated from:

$$RBPS = RBPD = \frac{R_{SHB} \cdot (L + D_{gg})}{2 \cdot W}$$

$$RBSB = \frac{(factor - even - odd) \cdot NF \cdot D_{SCB} \cdot R_{SHB}}{W}$$

$$RBDB = \frac{(factor - even - odd) \cdot NF \cdot D_{DCB} \cdot R_{SHB}}{W}$$

factor-even-odd = 0.5 for even number of fingers (NF)

factor-even-odd = 1 for odd number of fingers (NF)

The values of the elements of the SPICE equivalent circuit are calculated from device dimensions of the actual device (W, L, NF...) and additional model parameters like the gate sheet resistance RSHG.

Following is a SPICE netlist for the fully scalable BSIM3 model.

```
LINK CIRC Circuit
{
data
{
circuitdeck
{
*
* --------------------------------------------------------------------------------
* Fully scalable subcircuit model for BSIM3v3 RF n-type devices
* Simulator: UCB Spice3e2
* Model: BSIM3 Modeling Package
* Date: 08.11.2003
* Origin: ICCAP_ROOT/..../bsim3/circuits/spice3/cir/rf_nmos_scale.cir
* --------------------------------------------------------------------------------
*
.subckt bsim3_rf_extract 1 2 3 4
*
*--- Information for model implementation ------------------------------------------
*
* Due to the limitation of UCB spice3e2, the equations for the scaled RF model behavior are
included in the DUT parameters of the DUT RF_Transistor_Scale in the IC-CAP model
BSIM3_RF_Extract
*
* To implement this scalable model in your target simulator, please include those equations
using the appropriate syntax in the final model deck.
*
*--- BSIM3 model card ---------------------------------------------------------
#echo .MODEL BSIM3_HF NMOS
#echo + LEVEL=$mpar(LEVEL=8) VERSION=3.2.4 BINUNIT=$mpar(BINUNIT=2)
#echo + MOBMOD=$mpar(MOBMOD=1) CAPMOD=$mpar(CAPMOD=3) NOIMOD=$mpar(NOIMOD=1)
#echo + PARAMCHK= $mpar(PARAMCHK=1) DELTA=$mpar(DELTA=0.01) TNOM=$mpar(TNOM=27)
#echo + TOX=$mpar(TOX=7.5E-9) TOXM=$mpar(TOXM=7.5E-9)
#echo + NCH=$mpar(NCH=1.7e17) XJ=$mpar(XJ=1.5E-7) NGATE=$mpar(NGATE=0) RSH=$mpar(RSH=0)
#echo +
#echo + VTH0=$mpar(VTH0=0.7) K1=$mpar(K1=0.53) K2=$mpar(K2=-0.013) K3=$mpar(K3=0)
#echo + K3B=$mpar(K3B=0) W0=$mpar(W0=2.5E-6) NLX=$mpar(NLX=0.174u) DVT0=$mpar(DVT0=2.2)
#echo + DVT1=$mpar(DVT1=0.53) DVT2=$mpar(DVT2=-0.032) DVT0W=$mpar(DVT0W=0)
#echo + DVT1W=$mpar(DVT1W=5.3E6) DVT2W=$mpar(DVT2W=-0.032)
#echo + ETA0=$mpar(ETA0=0) ETAB=$mpar(ETAB=0) DSUB=$mpar(DSUB=0.56)
#echo +
#echo + U0=$mpar(U0=670) UA=$mpar(UA=2.25E-9) UB=$mpar(UB=5.87E-19) UC=$mpar(UC=4.65E-11)
#echo + VSAT=$mpar(VSAT=8e4) A0=$mpar(A0=1) AGS=$mpar(AGS=0) B0=$mpar(B0=0)
#echo + B1=$mpar(B1=0) KETA=$mpar(KETA=-0.047) A1=$mpar(A1=0) A2=$mpar(A2=1)
#echo + RDSW=$mpar(RDSW=0) PRWB=$mpar(PRWB=0) PRWG=$mpar(PRWG=0) WR=$mpar(WR=1)
#echo +
#echo + WINT=$mpar(WINT=0) WL=$mpar(WL=0) WLN=$mpar(WLN=1) WW=$mpar(WW=0)
#echo + WWN=$mpar(WWN=1) WWL=$mpar(WWL=0) DWG=$mpar(DWG=0) DWB=$mpar(DWB=0)
#echo + LINT=$mpar(LINT=0) LL=$mpar(LL=0) LLN=$mpar(LLN=1) LW=$mpar(LW=0)
#echo + LWN=$mpar(LWN=1) LWL=$mpar(LWL=0)
#echo + VOFF=$mpar(VOFF=-0.08) NFACTOR=$mpar(NFACTOR=1) CIT=$mpar(CIT=0)
#echo + CDSC=$mpar(CDSC=2.4E-4)
#echo + CDSCB=$mpar(CDSCB=0) CDSCD=$mpar(CDSCD=0) PCLM=$mpar(PCLM=1.3)
#echo +PDIBLC1=$mpar(PDIBLC1=0.39)
```

```
#echo + PDIBLC2=$mpar(PDIBLC2=0.0086) PDIBLCB=$mpar(PDIBLCB=0.0) DROUT=$mpar(DROUT=0.56)
#echo + PSCBE1=$mpar(PSCBE1=4.24E8)
#echo + PSCBE2=$mpar(PSCBE2=1.0E-5) PVAG=$mpar(PVAG=0) VBM=$mpar(VBM=-3)
#echo + ALPHA0=$mpar(ALPHA0=0) ALPHA1=$mpar(ALPHA1=0) BETA0=$mpar(BETA0=30)
#echo +
#echo + JS=1e-20 JSW=1.0E-20 NJ=1 IJTH=$mpar(IJTH=0.1)
#echo +
#echo + CJ=0 MJ=0.5 PB=1 CJSW=0
#echo + MJSW=0.33 PBSW=1 CJSWG=$mpar(CJSWG=5E-10) MJSWG=$mpar(MJSWG=0.33)
#echo + PBSWG=$mpar(PBSWG=1) CGDO=$mpar(CGDO=0) CGSO=$mpar(CGSO=0) CGBO=$mpar(CGBO=0)
#echo + CGSL=$mpar(CGSL=0) CGDL=$mpar(CGDL=0) CKAPPA=$mpar(CKAPPA=0.6) CF=$mpar(CF=0)
#echo + NOFF=$mpar(NOFF=1) VOFFCV=$mpar(VOFFCV=0) ACDE=$mpar(ACDE=1) MOIN=$mpar(MOIN=15)
#echo + DLC=$mpar(DLC=0) DWC=$mpar(DWC=0) LLC=$mpar(LLC=0) LWC=$mpar(LWC=0)
#echo + LWLC=$mpar(LWLC=0) WLC=$mpar(WLC=0) WWC=$mpar(WWC=0) WWLC=$mpar(WWLC=0)
#echo + CLC=$mpar(CLC=0.1E-6) CLE=$mpar(CLE=0.6)
#echo + ELM=$mpar(ELM=2) XPART=$mpar(XPART=0.5)
#echo +
#echo + KT1=$mpar(KT1=-0.11) KT1L=$mpar(KT1L=0) KT2=$mpar(KT2=0.022) UTE=$mpar(UTE=-1.5)
#echo + UA1=$mpar(UA1=4.31E-9) UB1=$mpar(UB1=-7.6E-18) UC1=$mpar(UC1=-5.6E-11)
#echo + AT=$mpar(AT=3.3E4)
#echo + PRT=$mpar(PRT=0) XTI=$mpar(XTI=3.0) TPB=$mpar(TPB=0) TPBSW=$mpar(TPBSW=0)
#echo + TPBSWG=$mpar(TPBSWG=0) TCJ=$mpar(TCJ=0) TCJSW=$mpar(TCJSW=0) TCJSWG=$mpar(TCJSWG=0)
#echo +
#echo + AF=$mpar(AF=1.5) EF=$mpar(EF=1.5) KF=$mpar(KF=1e-17) EM=$mpar(EM=4.1E7)
#echo + NOIA=$mpar(NOIA=2e29) NOIB=$mpar(NOIB=5e4) NOIC=$mpar(NOIC=-1.4e-12)
*
*
*--- Parasitic diode model cards ------------------------------------------------
#echo .MODEL bsim_diode_area D
#echo + CJO=$mpar(CJ=5E-4) VJ=$mpar(PB=1) M=$mpar(MJ=0.5)
#echo + IS=$mpar(JS=1.0E-4) N=$mpar(NJ=1)
*
#echo .MODEL bsim_diode_perim D
#echo + CJO=$mpar(CJSW=5E-10) VJ=$mpar(PBSW=1) M=$mpar(MJSW=0.33)
#echo + IS=$mpar(JSW=1.0E-12) N=$dpar(CALC.NJSW=1)
*
*
*--- Additional model parameters necessary for scalability---------------------------
* - scalable external capacitors and inductors to account for cross coupling in the metal
stripes and additional delay due to large sizes
* - a scalable substrate network
* - a scalable channel length reduction
*
#echo * CGDEXT0=$mpar(CGDEXT0=1e-9) ext. cap. gate-drain per gate width and finger [F/m]
#echo * CGSEXT0=$mpar(CGSEXT0=1e-9) ext. cap. gate-source per gate width and finger [F/m]
#echo * CDSEXT0=$mpar(CDSEXT0=1e-9) ext. cap. drain-source per gate width and finger [F/m]
#echo * RSHG=$mpar(RSHG=25) gate sheet resistance [Ohm sq]
#echo * LDRAIN0=$mpar(LDRAIN0=1e-6) drain inductance per gate width and gate finger [H/m]
#echo * LGATE0=$mpar(LGATE0=1e-6) gate inductance per gate width and gate finger [H/m]
```

```
#echo * LSOURCE0=$mpar(LSOURCE0=1e-6) source inductance per gate width and finger [H/m]
#echo * LBULK0=$mpar(LBULK0=1e-6) bulk inductance per gate width and finger [H/m]
#echo * RSHB=$mpar(RSHB=25) bulk sheet resistance [Ohm sq]
#echo * DSBC=$mpar(DSBC=2e-6) distance source implant to bulk contact [m]
#echo * DDBC=$mpar(DDBC=2e-6) distance drain implant to bulk contact [m]
#echo * DGG=$mpar(DGG=2e-6) distance gate to gate [m]
#echo * DL0=$mpar(DL0=0) basic channel length reduction correction [m]
#echo * DL1=$mpar(DL1=0) channel length reduction correction 1. and 2. outer fingers [m]
#echo * DL2=$mpar(DL2=0) channel length reduction correction outer fingers [m]
*
*
* --------- Gate network ----------------------------------------------------------
#echo LGATE 2 20 $dpar(CALC.LGATE=0.1p)
#echo RGATE 20 21 $dpar(CALC.RGATE=10)
#echo CGDEXT 20 10 $dpar(CALC.CGDEXT=0.1f)
#echo CGSEXT 20 30 $dpar(CALC.CGSEXT=0.1f)
*
* --------- Drain network ---------------------------------------------------------
#echo LDRAIN 1 10 $dpar(CALC.LDRAIN=0.1p)
#echo CDSEXT 10 30 $dpar(CALC.CDSEXT=0.1f)

* --------- Source network --------------------------------------------------------
#echo LSOURCE 3 30 $dpar(CALC.LSOURCE=0.1p)
*
* --------- Substrate network -----------------------------------------------------
* Diodes are for n-type MOS transistors
*
#echo Djdb_area 12 10 bsim_diode_area AREA=$dpar(x_rf_transistor.AD=10e-12)
#echo Djdb_perim 12 10 bsim_diode_perim AREA=$dpar(x_rf_transistor.PD=22e-6)
*
#echo Djsb_area 32 30 bsim_diode_area AREA=$dpar(x_rf_transistor.AS=10e-12)
#echo Djsb_perim 32 30 bsim_diode_perim AREA=$dpar(x_rf_transistor.PS=22e-6)
*
#echo RBDB 12 40 $dpar(CALC.RBDB=100)
#echo RBSB 32 40 $dpar(CALC.RBSB=100)
#echo RBPD 12 41 $dpar(CALC.RBPD=100)
#echo RBPS 32 41 $dpar(CALC.RBPS=100)
*
#echo LBULK 4 40 $dpar(CALC.LBULK=0.1p)
*
*--- call single MOSFET ----------------------------------------------------------
#echo MAIN 10 21 30 41 BSIM3_HF
#echo + L=$dpar(x_rf_transistor.L=1u) W=$dpar(x_rf_transistor.W=10e-6)
#echo + AD=$dpar(x_rf_transistor.AD=10e-12) AS=$dpar(x_rf_transistor.AS=10e-12)
#echo + PD=$dpar(x_rf_transistor.PD=22e-6) PS=$dpar(x_rf_transistor.PS=22e-6)
#echo + NRS=$dpar(x_rf_transistor.NRS=0) NRD=$dpar(x_rf_transistor.NRD=0)
#echo + NQSMOD=$mpar(NQSMOD=0)
*
.ends
}
}
}
```

Modeling Strategy

Modeling the AC behavior of a MOS device with the BSIM3v3 model heavily depends on the accurate modeling of the DC curves and the capacitances at low frequencies, for example, 10kHz to 1MHz. However, more and more applications, especially in the telecommunication industry, require the modeling of MOS transistors for the use in a frequency range of 1 to 10 GHz. Therefore, S-parameter measurements have to be done (see also "Test Structures for S-parameter Measurements" on page 299) to cover this frequency range by a proper device model.

As is pointed out, using the BSIM3v3 model for high frequency applications requires some special attention to the modeling strategy. We found the following procedure to give the most accurate results:

- Measurement of DC and CV curves.
- Extraction of the BSIM3v3 model parameters from DC and CV measurements with a special emphasis on a physically based extraction strategy. Here, model parameters should not be used for fitting purposes, they should have a correct physical meaning.
- The modeling of the output characteristic $I_d=f(V_{ds})$ and the output resistance $R_{out}=f(V_{ds})$ is very important for further S-parameter measurements (see Figure 123 and Figure 122).
- Performing S-parameter measurements and proper de-embedding of parasitics.
- The starting points of the S-parameter curves at the lowest frequency can be modeled by fitting the curves with DC and capacitance parameters. The following diagrams describe this influence on the high frequency behavior.

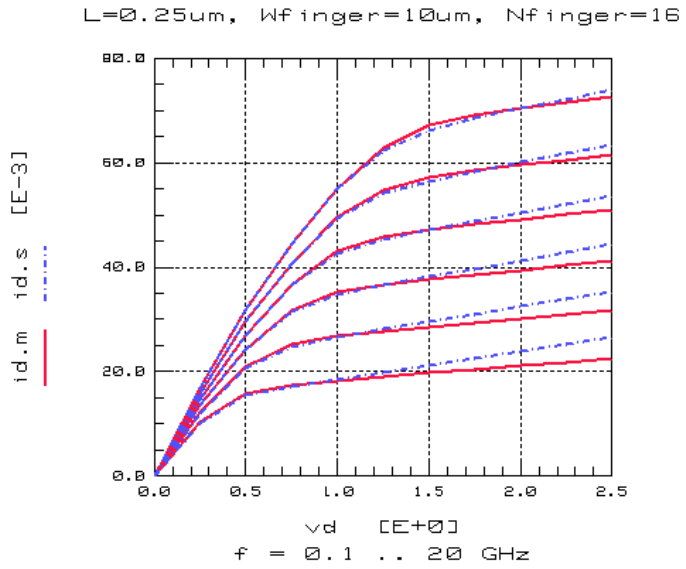


Figure 120 Incorrectly Modeled Drain Current

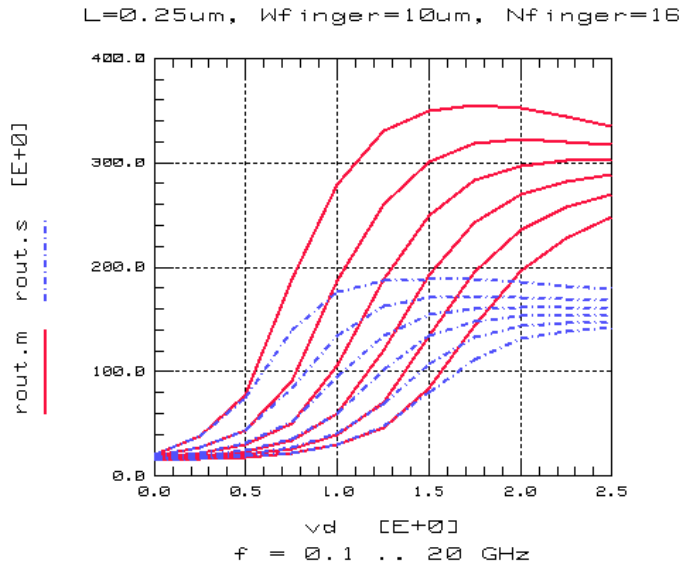


Figure 121 Incorrectly Modeled Output Characteristic

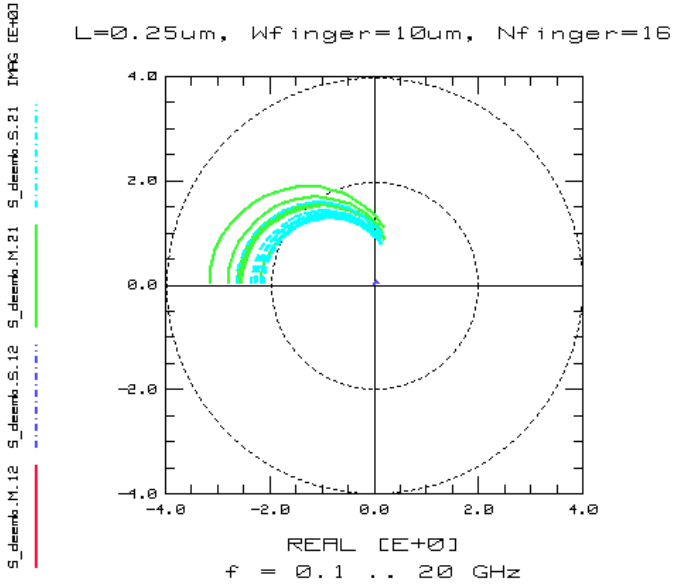


Figure 122 Influence of Incorrectly Modeled Output Characteristic on S21

- Extraction of the gate resistance from the input reflection S11 (see Figure 123)
- Verification of the gate-drain overlap capacitance for higher frequencies
- Extraction of the substrate resistance network parameters from S22 (see Figure 123)
- If a good fitting could not be found, additional peripheral elements like inductances at drain, gate, or source should be added in a further sub-circuit

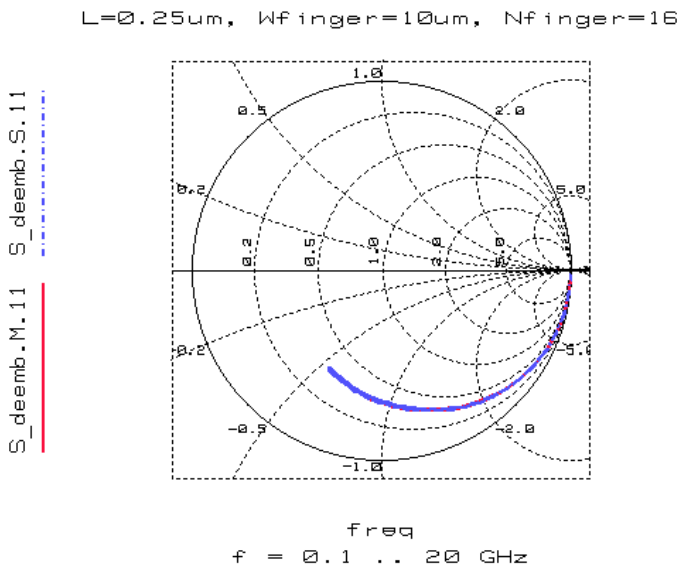


Figure 123 Input Reflection Parameter S11

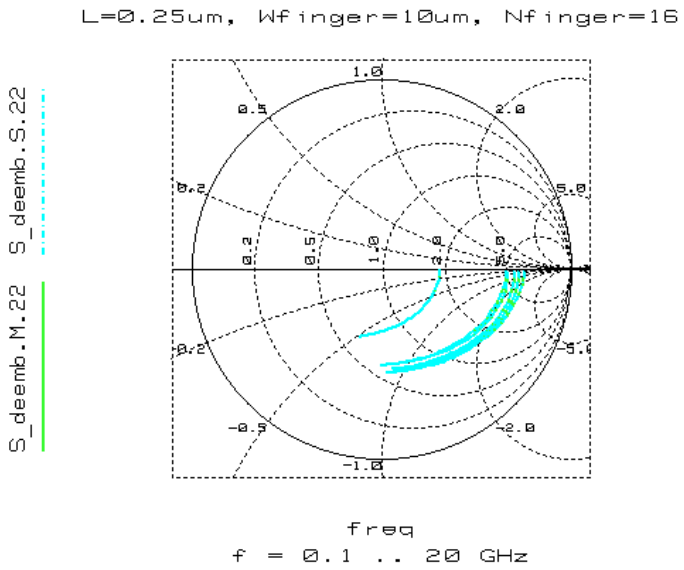


Figure 124 Output Reflection Parameter S22

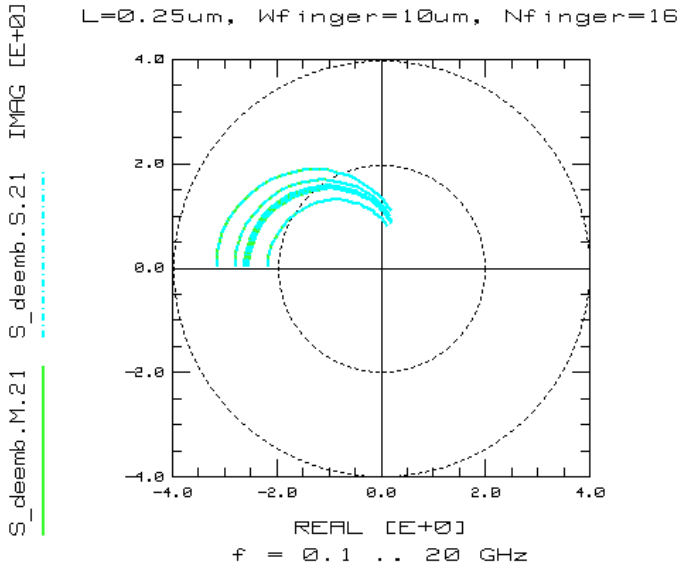


Figure 125 Forward Transmission Parameter S21

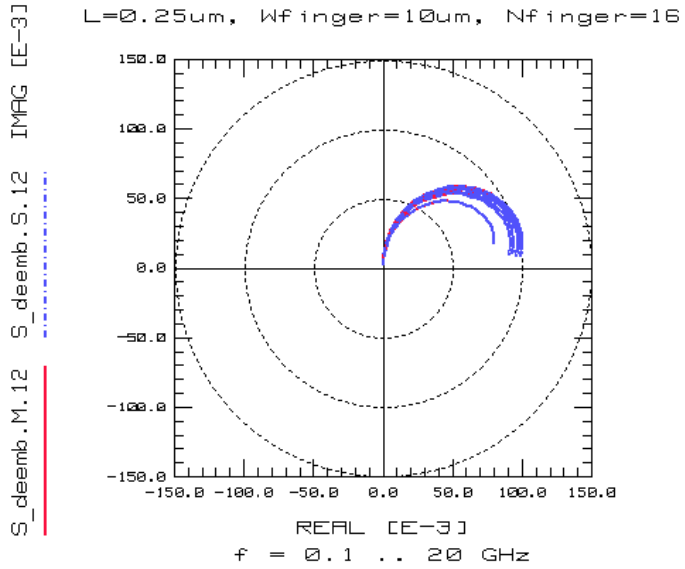


Figure 126 Reverse Transmission Parameter S12

Temperature Dependence

Please use the model *bsim3_tutor_temp.mdl* provided with the BSIM3v3 Modeling Package to visualize the temperature model parameters. Load the file into IC-CAP and run the different macros to see how certain parameters affect the device behavior of a deep submicron MOS transistor.

Built-in Temperature Dependencies

The BSIM3v3 model uses some physically based built-in temperature dependencies as listed below:

Temperature voltage:

$$V_{tm} = \frac{kBT}{q} \tag{70}$$

Intrinsic carrier concentration:

$$N_i = 2.6310^{16} T^{3/2} e^{\left(-\frac{6885}{T}\right)} \tag{71}$$

Unfortunately, the surface potential Φ_S, which is a very important model parameter from a physical point of view is **not** temperature dependent in BSIM3.

$$\Phi_s = 2V_{tm}(T_{nom}) ln\left(\frac{N_{ch}}{N_i(T_{nom})}\right) \tag{72}$$

Temperature Effects

In addition to the built-in temperature dependencies, the following temperature related effects are modeled in BSIM3. They are related to threshold voltage, mobility, saturation of carrier velocity, drain-source resistance, and the saturation current of the drain/source bulk diodes.

a) Threshold Voltage

$$V_{th}(T) = V_{th}(T_{nom}) + \left(KT_1 + \frac{KT_1 L}{L_{eff}} + KT_2 V_{bseff}\right)\left(\frac{T}{T_{nom}} - 1\right) \quad \textbf{(73)}$$

The behavior of the threshold voltage for a large and a short device is shown in Figure 127.

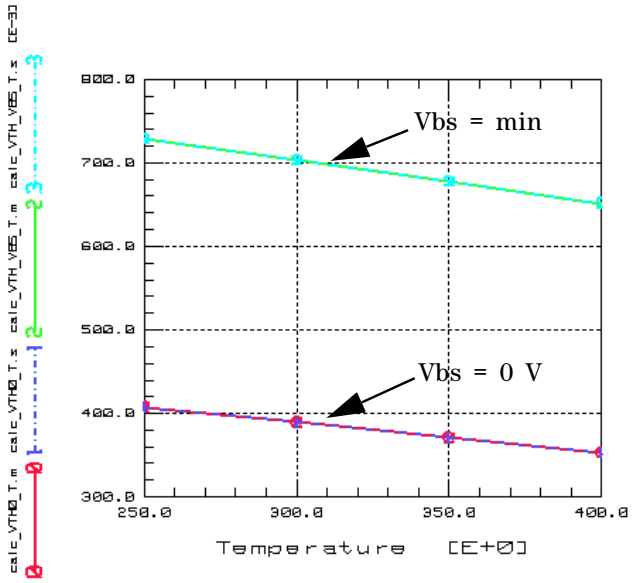


Figure 127 Threshold Voltage V_{th}=f(T) of a Large Device

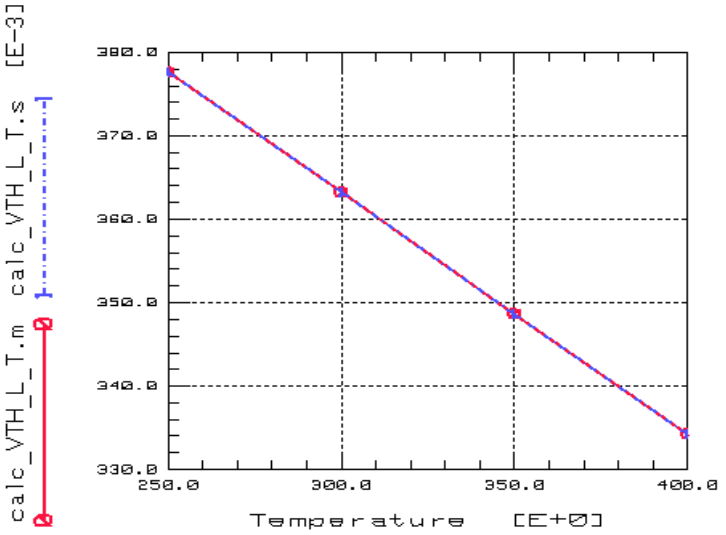


Figure 128 Threshold Voltage V_{th}=f(T) of a Short Device

b) Carrier Mobility

All four model parameters of the carrier mobility are implemented in BSIM3 with a temperature dependence:

$$\mu_0(T) = U0 \cdot \left(\frac{T}{T_{nom}}\right)^{UTE} \quad \textbf{(74)}$$

$$U_A(T) = UA + UA1\left(\frac{T}{T_{nom}} - 1\right) \quad \textbf{(75)}$$

$$U_B(T) = UB + UB1\left(\frac{T}{T_{nom}} - 1\right) \quad \textbf{(76)}$$

$$U_C(T) = UC + UC1\left(\frac{T}{T_{nom}} - 1\right) \quad \textbf{(77)}$$

The following two diagrams show the effect of temperature dependent mobility on the transconductance of a large transistor.

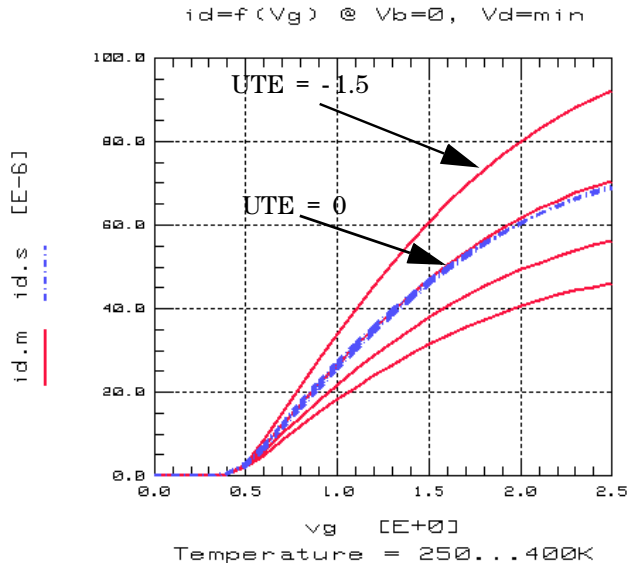


Figure 129 Temperature Dependence of Carrier Mobility U0: Influence on Drain Current

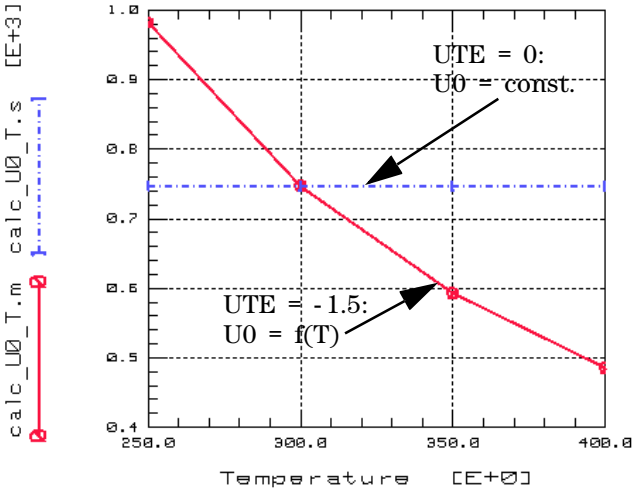


Figure 130 Temperature Dependence of Carrier Mobility U0: Dependence from UTE

c) Saturation of Carrier Velocity

The carrier velocity VSAT is reduced with increasing temperature as shown in the following equation and Figure 132:

$$V_{SAT}(T) = VSAT - AT\left(\frac{T}{T_{nom}} - 1\right) \quad \textbf{(78)}$$

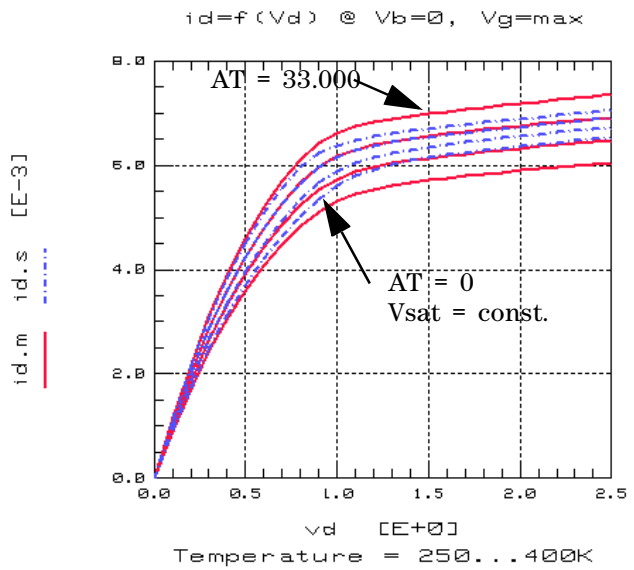


Figure 131 Output Characteristic $I_d=f(V_d,T)$

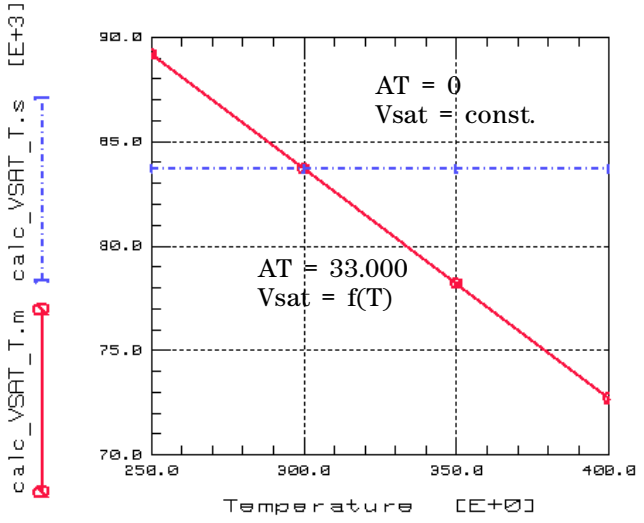


Figure 132 Output Characteristic $V_{SAT} = f(T)$

d) Drain source resistance

The temperature dependence of the drain source resistance is given by the following equation (see Figure 133):

$$R_{DSW}(T) = RDSW - PRT\left(\frac{T}{T_{nom}} - 1\right) \quad \textbf{(79)}$$

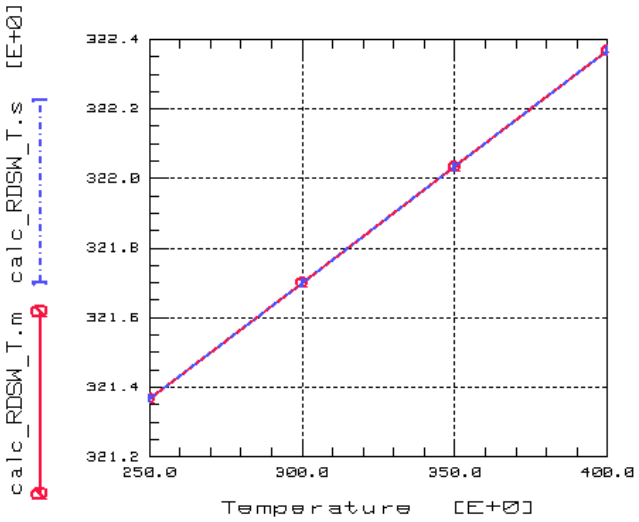


Figure 133 Drain source resistance R_{DSW} = f(T)

e) Saturation Current of Drain / Source Bulk Diodes

The temperature dependence of the drain/source bulk diodes is given by the following equation for the saturation current density J_S:

$$J_S(T) = JS \cdot e^{\left(\frac{\frac{E_{g0}}{V_{tm0}} - \frac{E_g}{V_{tm}} + XT\ ln\left(\frac{T}{T_{nom}}\right)}{N_J}\right)} \quad \textbf{(80)}$$

The influence of XTI on diode current and saturation current density J_S is shown below.

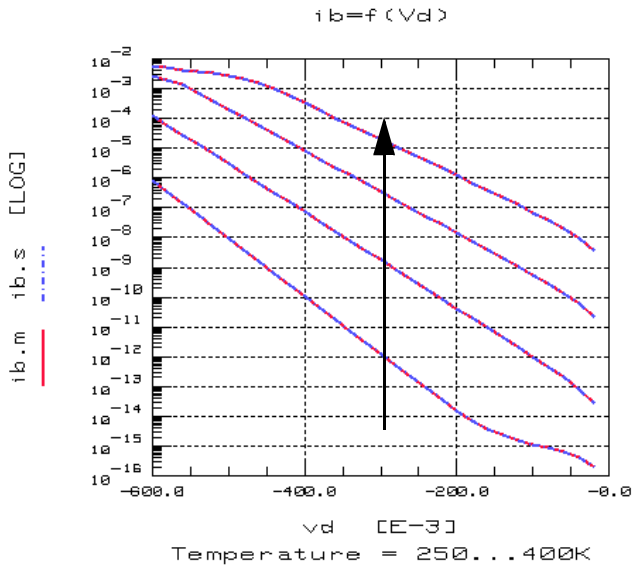


Figure 134 Saturation Current as Function of Temperature

Noise Model

There are two noise models implemented in BSIM3—a conventional noise model named Spice2 model and a newly formulated noise model, which is referred to as BSIM3v3 noise model. The following equations and diagrams should give insight into these two noise formulations.

Please use the model *bsim3_tutor_ac_noise.mdl* provided with the BSIM3v3 Modeling Package to visualize the model parameters. Load the file into IC-CAP and run the different macros to see how certain parameters affect the device behavior of a deep submicron MOS transistor.

Conventional Noise Model for MOS Devices

Flicker noise:

$$V_{noise,\,eff} = \frac{KF1_{ds}^{AF}}{C_{ox} \times L_{eff}^{2} f^{EF}} \tag{81}$$

Channel thermal noise:

$$V_{noise,\,eff} = \frac{8kT}{3}(g_m + g_{ds} + g_{mb}) \tag{82}$$

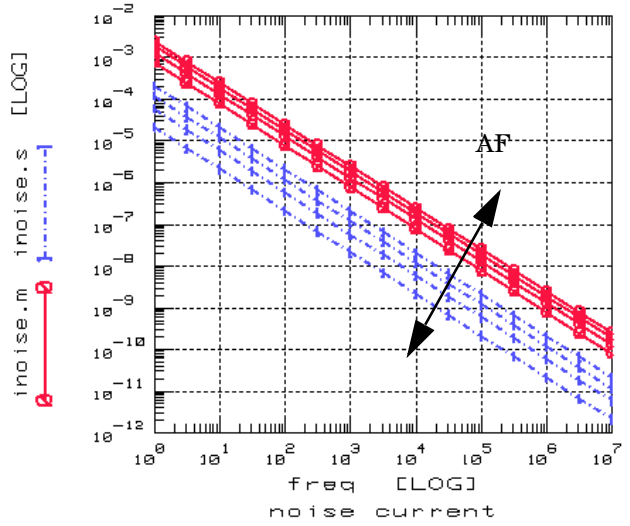


Figure 135 Influence of AF on Effective Noise Voltage

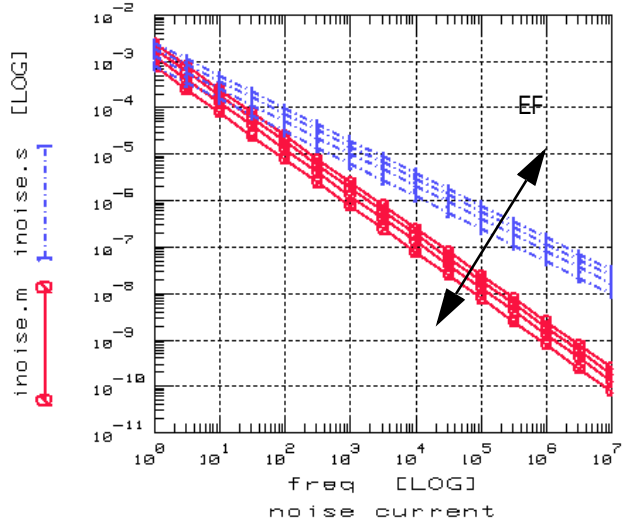


Figure 136 Influence of EF on Effective Noise Voltage

BSIM3v3 Noise Model

The BSIM3v3 noise model uses the following equation to describe the flicker noise:

(83)

$$V_{noise,\,eff} = \frac{V_{tm} q^2 l_{ds} \mu_{eff}}{10^8 C_{ox} L_{eff}^2 f^{EF}} \left[NOIA \log\left(\frac{No + 2 \times 10^{14}}{Nl + 2 \times 10^{14}} + NOIB(No - Nl) \right) + \ldots \right]$$

where:

No is the charge density at the source given by:

$$No = \frac{C_{ox}(V_{gs} - V_{th})}{q} \quad \textbf{(84)}$$

Nl is the charge density at the drain given by:

$$Nl = \frac{C_{ox}(V_{gs} - V_{th} - V_{ds}')}{q} \quad \textbf{(85)}$$

The channel thermal noise is given by:

$$V_{noise,\,eff} = \frac{4kT\mu_{eff}}{L_{eff}^2} |Q_{inv}| \quad \textbf{(86)}$$

with:

$$Q_{inv} = -W_{eff} L_{eff} C_{ox} V_{gsteff} \left(1 - \frac{A_{bulk}}{2(V_{gsteff} + 2V_{tm})} V_{dseff} \right) \quad \textbf{(87)}$$

SPICE Model Parameters

The model parameters of the BSIM3v3 model can be divided into several groups. The main model parameters are used to model the key physical effects in the DC and CV behavior of submicron MOS devices at room temperature. Here they are grouped into subsections related to the physical effects of the MOS transistor. The second group are the process related parameters. They should only be changed if a detailed knowledge of a certain MOS production process is given. The third group of parameters are the temperature modeling parameters. The following two groups are used to model the AC and noise behavior of the MOS transistor. Finally the last group contains flags to select certain modes of operations and user definable model parameters. For more details about these operation modes refer to the BSIM3v3 manual [1].

Main Model Parameters

Table 12 Main Model Parameters

Parameter	Description	Default Value (NMOS / PMOS)	Unit
Threshold Voltage			
VTHO	Ideal threshold voltage	0.7 / -0.7	V
K1	First-order body effect coefficient	0.5	$V^{0.5}$
K2	Second-order body effect coefficient	0.5	-
K3	Narrow width coefficient	80.0	-
K3B	Body effect coefficient of K3	0.0	1/V
W0	Narrow width parameter	2.5E-6	m
NLX	Lateral non-uniform doping coefficient	1.74E-7	m
VBM	Maximum applied body bias in VTH calculation	-5.0	V
DVT0	First coefficient of short-channel effect on VTH	2.2	-

Table 12 Main Model Parameters (continued)

Parameter	Description	Default Value (NMOS / PMOS)	Unit
DVT1	Second coefficient of short-channel effect on VTH	0.53	-
DVT2	Body-bias coefficient of short-channel effect on VTH	-0.032	1/V
DVT0W	First coefficient of narrow-channel effect on VTH	2.2	-
DVT1W	Second coefficient of narrow-channel effect on VTH	5.3E6	-
DVT2W	Body-bias coefficient of narrow-channel effect on VTH	-0.032	1/V
ETA0	DIBL coefficient in the subthreshold region	0.08	-
ETAB	Body-bias for the subthreshold DIBL effect	-0.07	1/V
DSUB	DIBL coefficient in subthreshold region	DROUT	-
Mobility			
U0	Mobility	670 / 250	$cm^2/(Vs)$
UA	First-order mobility degradation coefficient	2.25E-9	m/V
UB	Second-order mobility degradation coefficient	5.87E-19	$(m/V)^2$
UC	Body-effect of mobility degradation	-4.65E-11	$(m/V)^2$
Drain current			
VSAT	Saturation velocity	8.0E6	cm/s
A0	Bulk charge effect coefficient	1.0	-
A1	First non saturation factor	0/0.23	1/V
A2	Second non saturation factor	1.0/0.08	-
AGS	Gate-bias coefficient of Abulk	0.0	1/V
B0	Bulk charge effect coeff. for channel width	0.0	m
B1	Bulk charge effect width offset	0.0	m
KETA	Body-bias coefficient of the bulk charge effect.	-0.047	1/V
Subthreshold region			
VOFF	Offset voltage in the subthreshold region	-0.11	V
NFACTOR	Subthreshold swing factor	1.0	-

Table 12 Main Model Parameters (continued)

Parameter	Description	Default Value (NMOS / PMOS)	Unit
DVT1	Second coefficient of short-channel effect on VTH	0.53	-
DVT2	Body-bias coefficient of short-channel effect on VTH	-0.032	1/V
DVT0W	First coefficient of narrow-channel effect on VTH	2.2	-
DVT1W	Second coefficient of narrow-channel effect on VTH	5.3E6	-
DVT2W	Body-bias coefficient of narrow-channel effect on VTH	-0.032	1/V
ETA0	DIBL coefficient in the subthreshold region	0.08	-
ETAB	Body-bias for the subthreshold DIBL effect	-0.07	1/V
DSUB	DIBL coefficient in subthreshold region	DROUT	-
Mobility			
U0	Mobility	670 / 250	$cm^2/(Vs)$
UA	First-order mobility degradation coefficient	2.25E-9	m/V
UB	Second-order mobility degradation coefficient	5.87E-19	$(m/V)^2$
UC	Body-effect of mobility degradation	-4.65E-11	$(m/V)^2$
Drain current			
VSAT	Saturation velocity	8.0E6	cm/s
A0	Bulk charge effect coefficient	1.0	-
A1	First non saturation factor	0/0.23	1/V
A2	Second non saturation factor	1.0/0.08	-
AGS	Gate-bias coefficient of Abulk	0.0	1/V
B0	Bulk charge effect coeff. for channel width	0.0	m
B1	Bulk charge effect width offset	0.0	m
KETA	Body-bias coefficient of the bulk charge effect.	-0.047	1/V
Subthreshold region			
VOFF	Offset voltage in the subthreshold region	-0.11	V
NFACTOR	Subthreshold swing factor	1.0	-

Table 12 Main Model Parameters (continued)

Parameter	Description	Default Value (NMOS / PMOS)	Unit
CIT	Interface trap density	0	F/m^2
CDSC	Drain-Source to channel coupling capacitance	2.4E-4	F/m^2
CDSCB	Body-bias coefficient of CDSC	0	F/Vm^2
CDSCD	Drain-bias coefficient of CDSC	0	F/Vm^2
Drain-source resistance			
RDSW	Parasitic resistance per unit width	0	$\Omega\mu$ m
WR	Width offset from Weff for RDS calculation	1.0	-
PRWB	Body effect coefficient of RDSW	0	$V^{-0.5}$
PRWG	Gate bias effect coefficient of RDSW	0	1/V
Channel geometry			
WINT	Channel width reduction on one side	0	m
WL	Coeff. of length dependence for width offset	0	m
WLN	Power of length dependence for width offset	1	-
WW	Coeff. of width dependence for width offset	0	m
WWN	Power of width dependence for width offset	1	-
WWL	Coeff. of length and width cross term for width offset	0	m
LINT	Channel length reduction on one side	0	m
LL	Coeff. of length dependence for length offset	0	m
LLN	Power of length dependence for length offset	1	-
LW	Coeff. of width dependence for length offset	0	m
LWN	Power of width dependence for length offset	1	-
LWL	Coeff. of length and width cross term for length offset	0	m
DWG	Coefficient of Weff's gate dependence	0	m/V
DWB	Coefficient of Weff's substrate dependence	0	$m/V^{0.5}$
Output resistance			

Table 12 Main Model Parameters (continued)

Parameter	Description	Default Value (NMOS/PMOS)	Unit
PCLM	Channel length modulation coefficient	1.3	-
PDIBLC1	First output resistance DIBL effect	0.39	-
PDIBLC2	Second output resistance DIBL effect	0.0086	-
PDIBLCB	Body effect coefficient of output resistance DIBL effect	0	1/V
DROUT	L dependent coefficient of the DIBL effect in output resistance	0.56	
PSCBE1	First substrate current body-effect coefficient	4.24E8	V/m
PSCBE2	Second substrate current body-effect coefficient	1.0E-5	m/V
PVAG	Gate dependence of Early voltage	0	-
ALPHA0	The first parameter of impact ionization	0	m/V
ALPHA1	Length dependent substrate current parameter	0	1/V
BETA0	The second parameter of impact ionization	30	
Diode characteristic			
JS	Source drain junction saturation density	1E-4	A/m^2
JSSW	Side wall saturation current density	0	A/m
NJ	Emission coefficient of junction	1	-
IJTH	Diode limiting current	0.1	A
Capacitance			
CJ	Source/drain bottom junction capacitance per unit area	5.0E-4	F/m^2
CJSW	Source/drain side junction capacitance per unit length	5.0E-10	F/m
CJSWG	Source/drain gate side junction capacitance per unit length	CJSW	F/m
MJ	Bottom junction capacitance grading coefficient	0.5	-
MJSW	Source/drain side junction capacitance grading coefficient	0.33	-
MJSWG	Source/drain gate side junction cap. grading coefficient	MJSW	-
PB	Bottom junction built-in potential	1.0	V
PBSW	Source/drain side junction built-in potential	1.0	V

Table 12 Main Model Parameters (continued)

Parameter	Description	Default Value (NMOS/PMOS)	Unit
PBSWG	Source/drain gate side junction built-in potential	PBSW	V
CGSO	Gate-source overlap capacitance per unit W	XJ*COX/2	F/m
CGDO	Gate-drain overlap capacitance per unit W	XJ*COX/2	F/m
GGBO	Gate-bulk overlap capacitance per unit W	0.0	F/m
CGSL	Light doped source-gate region overlap capacitance	0.0	F/m
CGDL	Light doped drain-gate region overlap capacitance	0.0	F/m
CKAPPA	Coefficient for lightly doped region overlap	0.6	F/m
CF	Fringing field capacitance	-	F/m
CLC	Constant term for the short channel model	0.1E-6	m
CLE	Exponential term for the short channel model	0.6	
DLC	Length offset fitting parameter from C-V	LINT	m
DWC	Width offset fitting parameter from C-V	WINT	m
NOFF	Subthreshold swing factor for CV model	1	-
VOFFCV	Offset voltage for CV model	0	V

Process Related Parameters

Table 13 Process Related Parameters

Parameter	Description	Default Value	Unit
TOXM	Gate oxide thickness at which parameters are extracted	15e-9	m
TOX	Gate oxide thickness	15E-9	m
XJ	Junction depth	150E-9	m
NCH	Doping concentration near interface	1.7E17	$1/cm^3$
NSUB	Doping concentration away from interface	6E16	$1/cm^3$
NGATE	Poly gate doping concentration	0	$1/cm^3$
VFB	Flat-band voltage	-1.0	V
gamma1	Body-effect near interface	$\gamma_1 = \frac{\sqrt{2q\varepsilon_{Si}N_{Ch}}}{}$	$V^{1/2}$
gamma2	Body-effect far from interface	γ_1	$V^{1/2}$
XT	Doping depth	1.55E-7	V
RSH	Source/Drain Sheet resistance	0	$\frac{\Omega}{square}$

Temperature Modeling Parameters

Table 14 Temperature Modeling Parameters

Parameter	Description	Default Value	Unit
UTE	Mobility temperature coefficient	-1.5	-
KT1	Threshold voltage temperature coefficient	-0.11	V
KT1L	Channel length dependence of KT1	0.0	Vm
KT2	Threshold voltage temperature coefficient	0.022	-
UA1	Temperature coefficient for UA	4.31E-19	m/V
UB1	Temperature coefficient for UB	-7.61E-18	$(m/V)^2$
UC1	Temperature coefficient for UC	-0.056	m/V^2
PRT	Temperature coefficient for RDSW	0.0	Ωμ m
AT	Saturation velocity temperature coefficient	3.3E4	m/s
XTI	Junction current temperature exponent coefficient	3.0	-
TPB	Temperature coefficient for PB	0	V/K
TPBSW	Temperature coefficient for PBSW	0	V/K
TPBSWG	Temperature coefficient for PBSWG	0	V/K
TCJ	Temperature coefficient for CJ	0	1/K
TCJSW	Temperature coefficient for CJSW	0	1/K
TCJSWG	Temperature coefficient for CJSWG	0	1/K

Flicker Noise Model Parameters

Table 15 Flicker Noise Model Parameters

Parameter	Description	Default Value (NMOS/PMOS)	Unit
NOIA	Noise parameter A	1E20 / 9.9E18	-
NOIB	Noise parameter B	5E4 / 2.4E3	-

Table 15 Flicker Noise Model Parameters (continued)

Parameter	Description	Default Value (NMOS / PMOS)	Unit
NOIC	Noise parameter C	-1.4E-12 / 1.4E12	-
EM	Saturation field	4.1E7	V/m
AF	Frequency exponent	1	-
EF	Flicker exponent	1	-
KF	Flicker noise parameter	0	-

Non-Quasi-Static Model Parameters

Table 16 Non-Quasi-Static Model Parameter

Parameter	Description	Default Value	Unit
ELM	Elmore constant of the channel	5	-

Model Selection Flags

Table 17 Model Selection Flags

Parameter	Value	Type of Model
LEVEL	8	BSIM3v3 model selector (in UCB SPICE)
MOBMOD	1	Mobility model
	2	
	3	
CAPMOD	0	Capacitance model
	1	
	2	
	3	
NQSMOD	0	Non quasi static model

Table 17 Model Selection Flags (continued)

Parameter	Value	Type of Model
	1	
NOIMOD	1	Noise model
	2	
	3	
	4	

User Definable Parameters

Table 18 User Definable Parameters

Parameter	Description	Default Value	Unit
XPART	Charge partitioning coefficient	0	-
DELTA	Parameter for smoothness of effective V_{ds} calculation	0.01	-

Additional Parameters needed for accurate RF modeling

Table 19 RF Parameters for the RF subcircuit

Parameter	Description	Default Value	Unit
RSHB	bulk sheet resistance	25	$\frac{\Omega}{square}$
DGG	distance between gate stripes	2E-6	m
DSCB	distance source to bulk contact	2E-6	m
DDCB	distance drain to bulk contact	2E-6	m
RBDB	resistance between bulk connection point and drain	100	Ω
RBSB	resistance between bulk connection point and source	100	Ω
RBPD	resistance between the region below the channel and the drain region	100	Ω

Table 19 RF Parameters for the RF subcircuit (continued)

Parameter	Description	Default Value	Unit
RBPS	resistance between the region below the channel and the source region	100	Ω

Test structures for Deep Submicron CMOS Processes

A very important prerequisite for a proper model parameter extraction is the selection of appropriate test structures. The following sections describe the necessary test structures for the determination of CV and DC model parameters. A very detailed description of ideal test structures can be found in the JESSI AC-41 reports [2].

Transistors for DC measurements

The minimum set of devices for a proper extraction of DC model parameters is marked with ■ in Figure 137. This means one transistor with large and wide channel (and therefore showing no short/narrow effects), one transistor with a narrow channel, one transistor with a short channel, and one device with both short and narrow channel. Please note that with this minimum set of devices some parameters cannot be determined correctly (see the chapter “Extraction of parameters”) and they are set to default values during the extraction. For an extraction of all model parameters and a better fit of the simulated devices over the whole range of designed gate length and gate width, use more devices with different gate lengths and gate widths as shown in Figure 137 with ○ signs. You can use additional devices, for example, for evaluating the extraction results for certain channel lengths and widths used in your process. They are marked ▲.

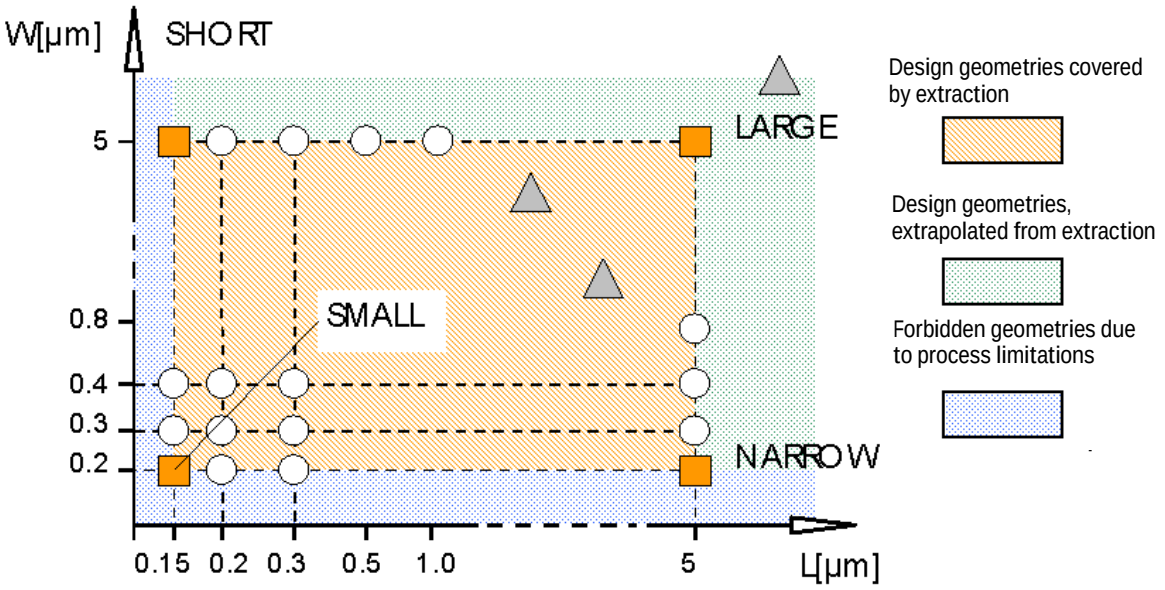


Figure 137 Recommended Test Transistor Geometries for proper parameter extraction

a) Requirements for Devices

Large For a proper extraction of the basic model parameters, the short and narrow channel effects should not affect the large device extraction. Also the drain-source-resistance parameters should not have an influence on the simulated behavior of the large device. For a typical 0.5 micron CMOS process with a gate oxide thickness of 11 nm, a large device with channel length of 10 microns and channel width of 10 microns was found to meet these requirements.

You can check this prerequisite if you only extract the parameters in the idvg/Large setup and then perform a simulation of the setup idvg/Large_m. After that simulation, perform the other geometry extractions and re-simulate the idvg/Large setup again. Now, the curve ID = f(V_{gs}) should not change more than roughly 5%

compared to the first simulation. If the difference is bigger, a larger device should be used to enable a good extraction of the basic model parameters.

Narrow For the DUT *Narrow_m* you should use a device with the smallest designed gate width of your process. Using more narrow devices will increase the number of parameters that can be extracted and will lead to a better fit of the curves over the range of different channel widths.

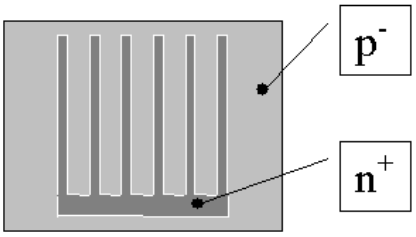
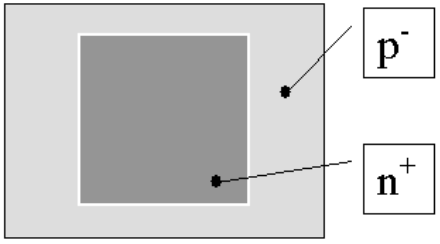
Short For the DUT *Short_m* you should use a device with the shortest designed gate length of your process. Using more short devices will increase the number of parameters that can be extracted and will lead to a better fit of the curves over the range of different channel lengths.

Small For the DUT *Small_m* you should use a device with the shortest designed gate length and the smallest designed gate width of your process. This small device will incorporate all short and narrow channel effects and will be an indicator of how good your parameter extractions are.

In general It is recommended to use the designed gate lengths and widths. Effects due to under diffusion or decrease of poly-Si gate length are sufficiently covered by the extraction routines and the model itself.

Drain/Source – Bulk Diodes for DC Measurements

Table 20 Test Structures for Drain/Source - Bulk Diodes

DUT	Shape	Comment
Diode_Perim_m		Finger diode with a large perimeter and a small area (shown here for an n-type device)
Diode_Area_m		Area diode with a large area and a small perimeter (shown here for an n-type device)

Test Structures for CV Measurements

Table 21 Test Structures for CV Measurements

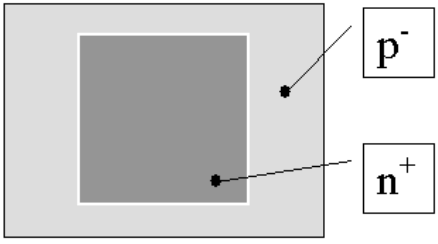
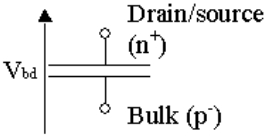
DUT	Shape	Applied bias (n-type)	Comment
C_Area_m (pn-junction)			Area diode with a large area, a small perimeter and the doping concentration n^+ of the drain/source region (shown here for an n-type device).

Table 21 Test Structures for CV Measurements

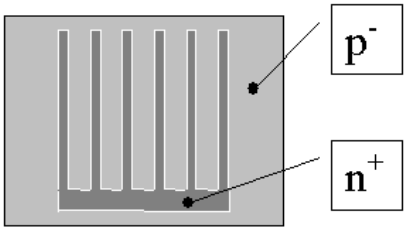
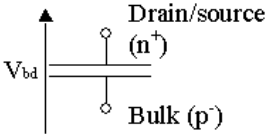
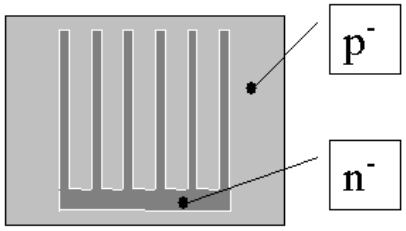
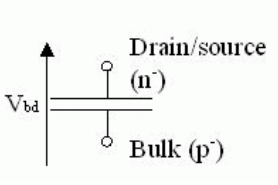
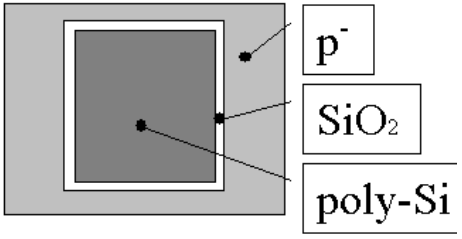
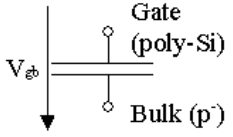
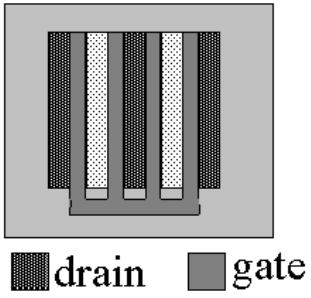
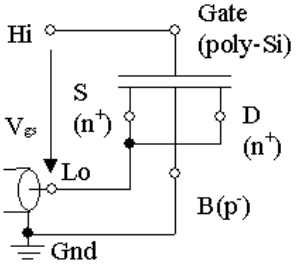
DUT	Shape	Applied bias (n-type)	Comment
C_Perim_m (pn-junction)			Finger diode with a large perimeter, a small area and the doping concentration n^+ of the drain/source region (shown here for an n-type device).
C_Perim_Gate_m (pn-junction)			Finger diode with a large perimeter, a small area and the doping concentration n^- of the LDD region (shown here for an n-type device).
C_Oxide_m (Gate oxide)			Large area MOS capacitor
C_Gate_SD_m (Overlap gate - drain/source)			A large number of parallel switched LDD MOS transistors (e.g., 200 transistors with L=0.25µm, W=10.0µm) or multifinger transistors (see shape)

Table 21 Test Structures for CV Measurements

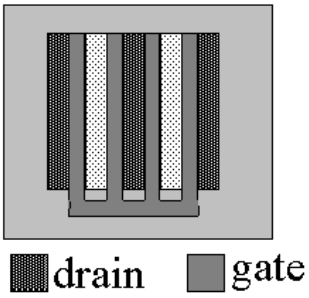
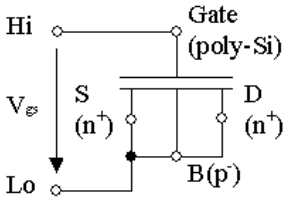
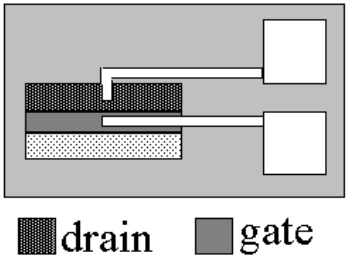
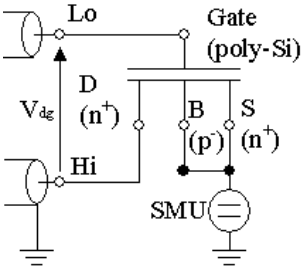
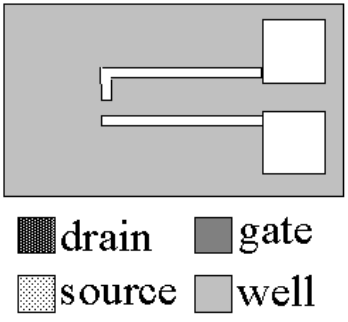
DUT	Shape	Applied bias (n-type)	Comment
C_Gate_SDB_m (Overlap gate - bulk/drain/ source)			A large number of parallel switched LDD MOS transistors (e.g., 200 transistors with L=0.25µm, W=10.0µm) or multifinger transistors (see shape)

Table 22 Test Structures for Intrinsic Capacitance Measurements

DUT	Shape	Applied bias (n-type)	Comment
C_Gate_D_m (Overlap gate - drain with applied DC bias)			A short channel transistor, with such a channel width or different fingers that the measurement instrument (CV-meter or Network Analyzer) is not overloaded by DC currents and a reasonable capacitance value can be measured.
OPEN		———	For very small capacitance values, an additional OPEN calibration structure on chip is necessary to compensate the capacitance of pads and lines to the transistor.

Testchips

You will find an example for a test chip design, which meets most of the requirements of the extraction of BSIM3v3 model parameter, in the JESSI Report AC 41 94-3 “Description of parametrized European Mini Test Chip.” Please check also the test chip design of the Fabless Semiconductor Association in the U.S. (http://www.fsa.org).

Test Structures for S-parameter Measurements

a) Test Structures

Performing S-parameter measurements with MOS devices on a wafer requires properly designed test structures that meet certain requirements:

- The test devices must drive enough current for correct measurement results
- They should fulfill the specifications for high frequency probes
- Additional structures should be available for the measurement of parasitic elements to de-embed them from the measurements on the test device

A principle layout of such a test structure is shown in Figure 138 [9].

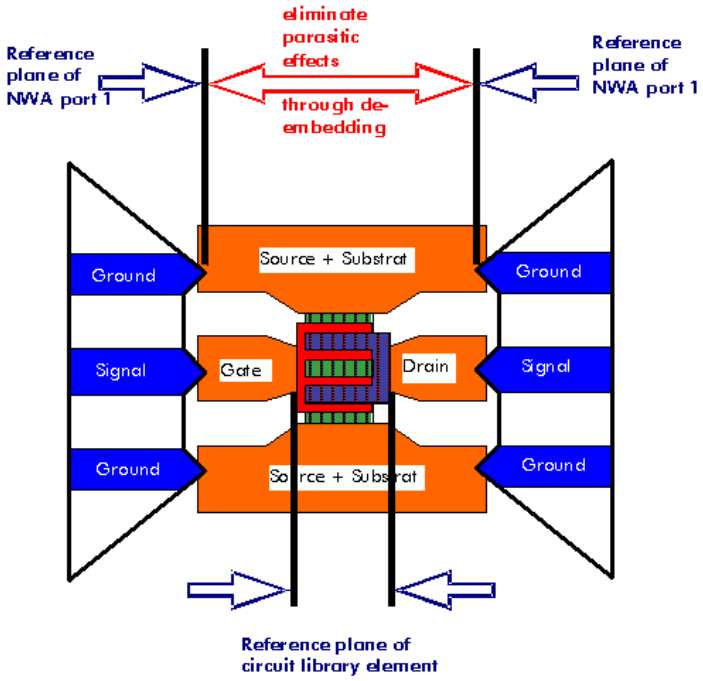


Figure 138 Layout of a Test Structure for a MOS Transistor

The MOS transistor is designed as a finger structure with four common gates, three source areas and two drain areas. In summary, this compact layout results in a very wide gate width, which can drive a high current I_{ds}.

The probes are connected in a Ground-Signal-Ground scheme according to the recommendations in [4]. As it is shown above, the calibration plane of the network analyzer is at the end of the probe head. This means, the transmission lines that connect the DUT with the probe head must be modeled and their effect must be de-embedded from the measured data of the DUT. This can be done by measuring an OPEN and a SHORT test device without a DUT and using these measurements to de-embed the parasitic influence of the pads. The following two figures show the design of these OPEN and SHORT test structures. Both of these test structures will be used for a simple and effective

de-embedding procedure (OPEN_SHORT) as will be shown later. Additional test devices, like a THROUGH device can be used to verify the de-embedding strategy. In general, the complexity of the de-embedding procedure depends on the frequency range of the measurements and the design of the test structures. However, a proper de-embedding is the absolute pre-requisite for an accurate AC modeling of the MOS transistor.

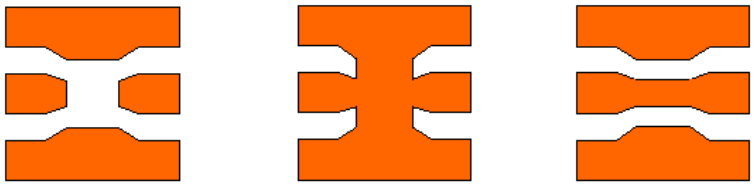


Figure 139 OPEN, SHORT and THROUGH structure without MOS transistor

Table 23 Test Structures for S-parameter Measurements

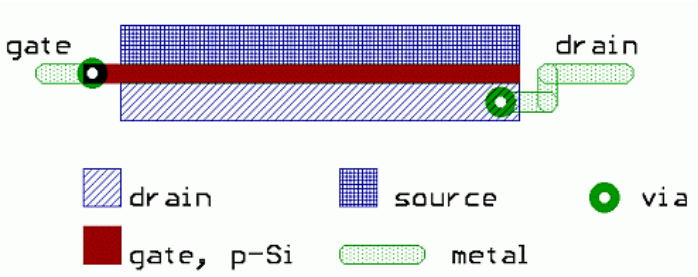
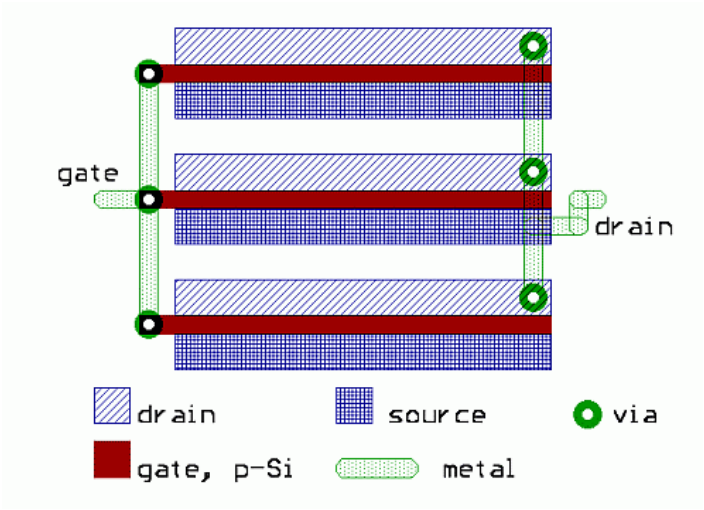
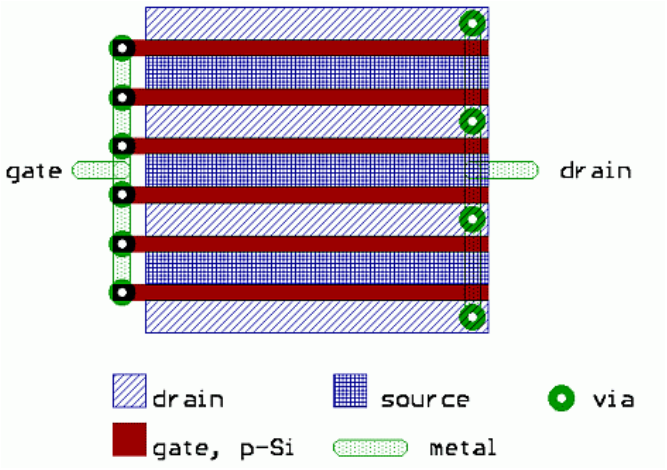
Test Structure	Top View	Input in '___Define DUT	
One single transistor		No of gates: No of drains: No of sources: L: W: Area drain: Area source: Per. drain: Per. source:	1 1 1 L W AD AS PD PS

Table 23 Test Structures for S-parameter Measurements

Test Structure	Top View	Input in '___Define DUT
n parallel transistors		No of gates: 3 No of drains: 3 No of sources: 3 L: L W: W [1] Area drain: AD [1] Area source: AS[1] Per. drain: PD[1] Per. source: PS[1]
multifinger transistors		No of gates: 6 No of drains: 4 No of sources: 3 L: L W: W[1] Area drain: AD[1] Area source: AS[1] Per. drain: PD[1] Per. source: PS[1]

b) De-embedding procedures

The DUT *Deembedding* > *Calculation* contains five different setups, two for general purposes and three with different de-embedding methods, to be selected depending on the availability of test structures and the frequency range of measurements. They are:

- no_deembedding
- resetDeembedding

and

- deembed_open
- deembed_open_short
- deembed_user_defined

1. OPEN:

This the simplest way of de-embedding and is often used for frequency ranges up to 10 GHz. It is assumed that the parasitics can be modeled using the following equivalent circuit:

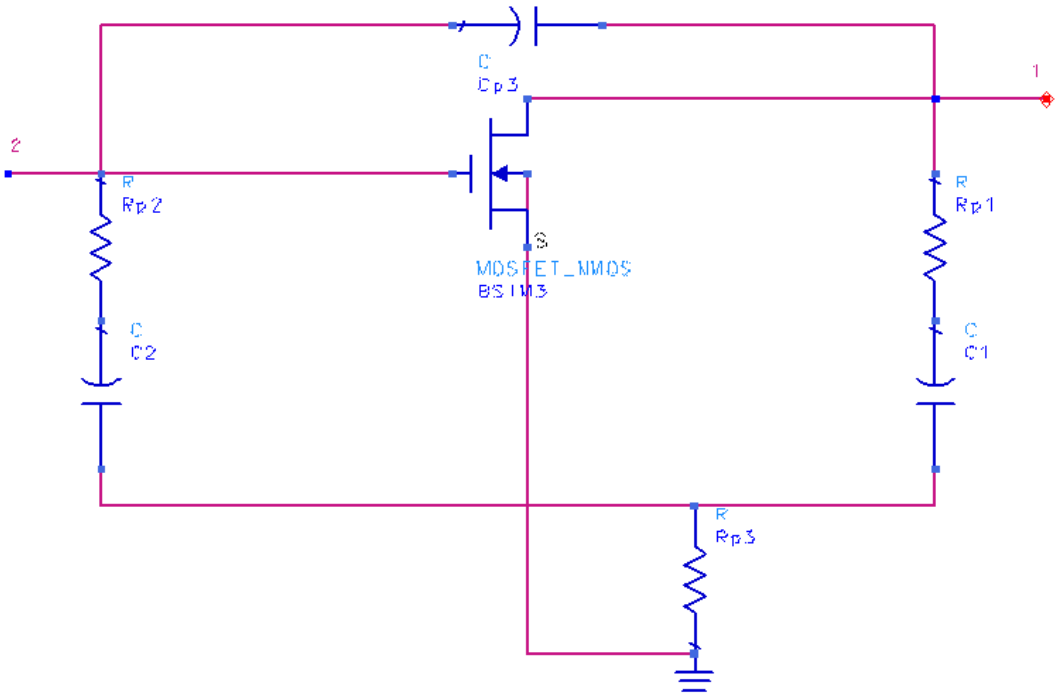
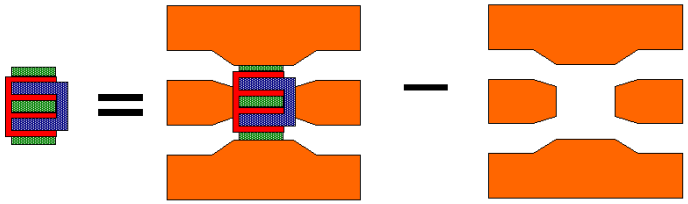


Figure 140 Equivalent circuit for the parasitic elements (including MOS-Transistor)

The OPEN device is measured and the S-parameters of the DUT are calculated as shown next:

S_{total} -> Y_{total}, S_{open} -> Y_{open}



$$Y_{DUT} = Y_{Total} - Y_{OPEN}$$

Y_{dut} -> S_{dut}

where:

S_{total} -> measured S-parameters of the DUT including parasitics

S_{open} -> measured S-parameters of the OPEN test structure

S_{dut} -> S-parameters of the DUT without influence of the parasitics

Yxxx -> transformed Y-parameters with:

'Y_{total} = TwoPort(S_{total},"S","Y")'

The typical behavior of this test structure is shown in Figure 141 and Figure 142.

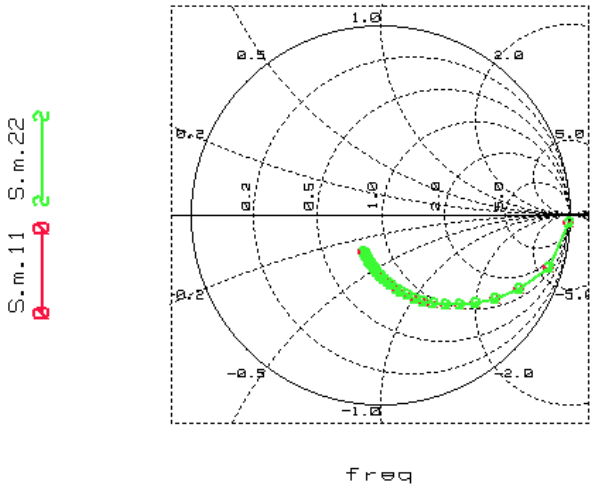


Figure 141 $S_{11,22}$ of the OPEN structure

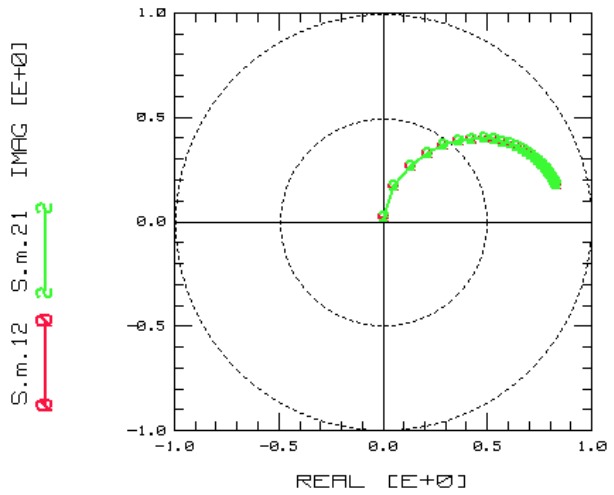


Figure 142 $S_{12,21}$ of the OPEN structure

2.OPEN_SHORT:

This is a very fast and effective way of de-embedding from measurements of an OPEN and a SHORT device. It is useful for frequencies above roughly 3.5 GHz if the accuracy of the OPEN method is not satisfying.

This method is described in detail in the IC-CAP demo_features. (See the file: *$ICCAP_ROOT/examples/demo_features/4extraction/deemb_short_open.mdl*)

It is assumed that the parasitics can be modeled using the following equivalent circuit:

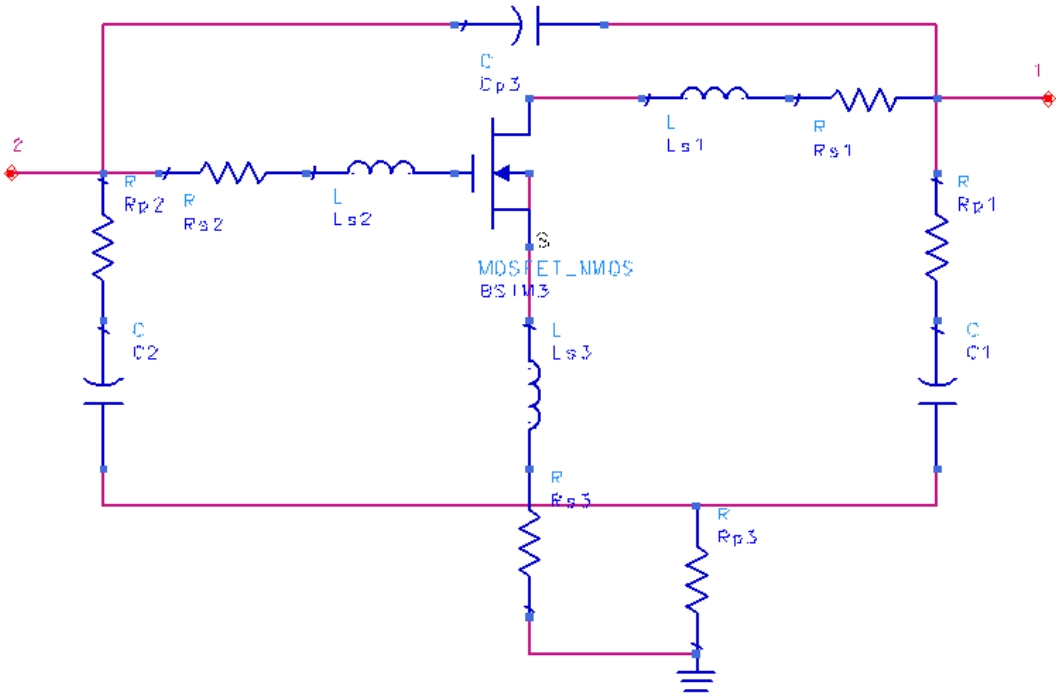


Figure 143 Detailed equivalent circuit of MOS-Transistor

The transistor is located between nodes: Gate = 222, Drain = 111, Source, Bulk = 333

Regarding the two test structures OPEN and SHORT and their equivalent circuits, it is assumed that there are ONLY parallel parasitics followed by serial parasitics. If this pre-requisite is valid, the measured data of the SHORT device and the measured data from the DUT have to be de-embedded from the outer parallel parasitic elements first (after a conversion of S to Y parameters):

$$Z_{dut\_without\_open} = Z(Y_{total} - Y_{open})$$

$$Z_{short\_without\_open} = Z(Y_{short} - Y_{open})$$

The subsequent step is to de-embed the measured data of the DUT from the serial parasitic elements and convert them back to S-parameters:

$$S_{dut} = S(Z_{dut\_without\_open} - Z_{short\_without\_open})$$

The typical behavior of the OPEN_SHORT structure is shown in the two figures below:

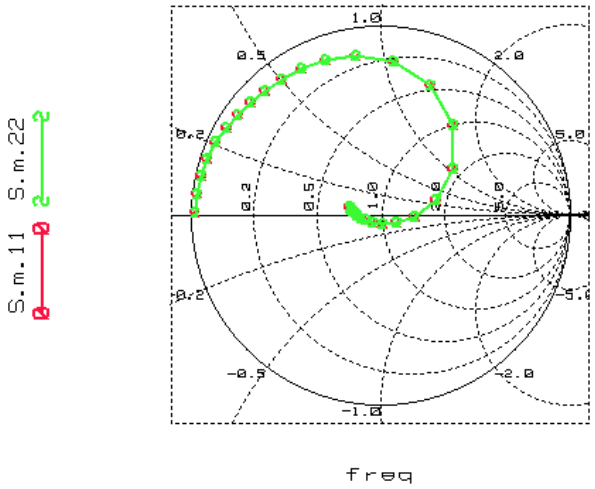


Figure 144 $S_{11,22}$ of the OPEN_SHORT structure

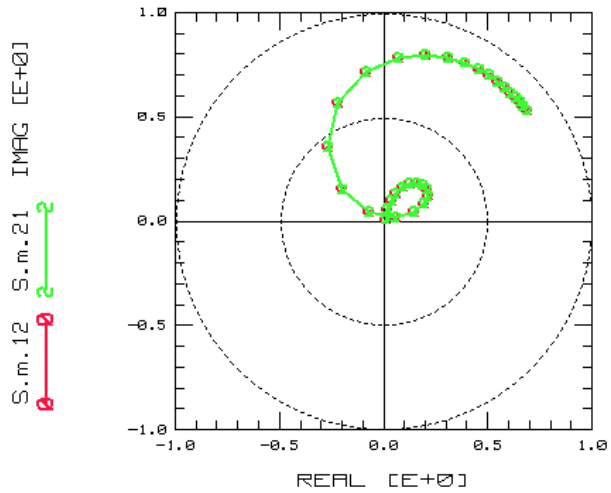


Figure 145 $S_{12,21}$ of the OPEN_SHORT structure

3.USER_DEFINED:

This setup can be used to implement user-specific de-embedding procedures with other test structures than OPEN and SHORT or to achieve a higher quality in de-embedding.

Please see the transform *deembed_all* to locate the entry point for your specific de-embedding procedure.

The ultimate tool for de-embedding with IC-CAP is the *De-embedding Tool-kit* where a large number of ready-to-go solutions together with the theoretical background can be found. Please contact Dr. Franz Sischka from Agilent EEsof (franz_sischka@agilent.com) for more details.

c) Verification procedures

The BSIM3v3 Modeling Package provides a method to verify the de-embedding. It uses a THROUGH dummy test device. After a correct de-embedding of the parasitic components, the S-parameters of the THROUGH should show the behavior

of an ideal, matched transmission line with Z0=50 Ohm and a TD that represents the electrical length of the through line in the THROUGH dummy device.

The S_{11} and S_{22} curve should be concentrated in the center of the Smith chart, while S_{21} and S_{12} should both begin at (1+j*0) and turn clockwise on the unity circle.

If these pre assumptions are not given, the following items should be checked:

- Is the calibration OK?
- If the OPEN method is used, consider to enhance the de-embedding quality by using the OPEN_SHORT method, which removes the inductive parasitics in the measured data.
- If the OPEN_SHORT method is used and the frequency is very high (>30 GHz), it should be checked whether the assumptions for using OPEN_SHORT are still given. The easiest way to do this is to model the OPEN and the SHORT device using the equivalent circuits given in Test_open and Test_short.

Extraction of Model Parameters

This section describes the parameter extraction sequence and the extraction strategy.

Parameter Extraction Sequence

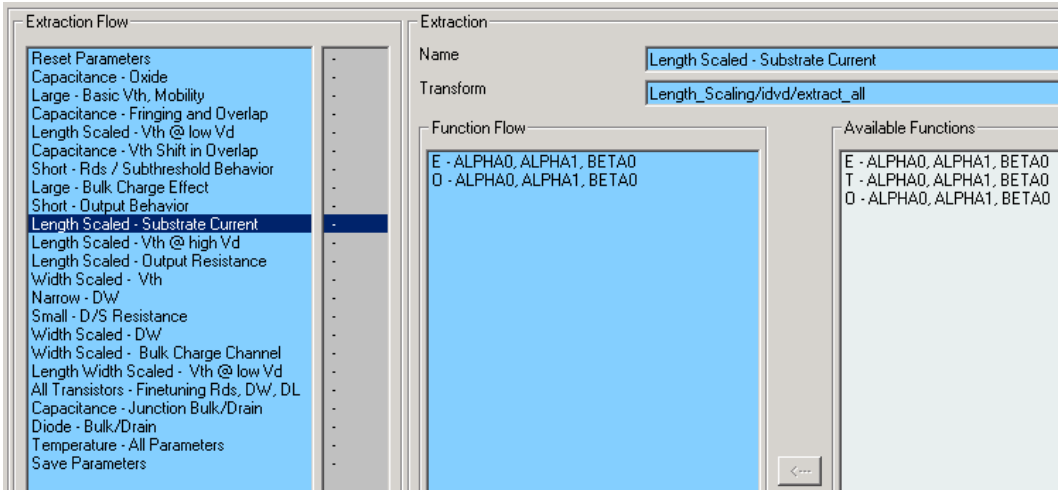
The default setting of the extraction flow is programmed according to a procedure found to give best extraction results. Using this macro, everything is done automatically.

The extraction functions are equipped with error and plausibility checks. If an error occurs or some parameters have strange or unrealistic values, you will get an error warning at the end of the macro.

In some cases, it can be useful not to extract all the parameters. For instance, if a 3.0 micron CMOS process has to be modeled with BSIM3, the typical short channel effects of the threshold voltage are not given and the extraction of the parameters DVT0, DVT1, and so on can result in very unrealistic values. In this case, those parameters should be removed from the extraction flow.

In general, if the macro produces errors, you should add the visual tuning feature to those parameters that caused the error. In a further run, the correspondent curves are simulated and displayed, and the user can try to find the source of the error.

The sequence of the model parameter extraction is shown in the following figure. You can modify this extraction flow by editing the flow if you find another sequence that better fits your special process.


Extraction Flow
Reset Parameters
Capacitance - Oxide
Large - Basic Vth, Mobility
Capacitance - Fringing and Overlap
Length Scaled - Vth @ low Vd
Capacitance - Vth Shift in Overlap
Short - Rds / Subthreshold Behavior
Large - Bulk Charge Effect
Short - Output Behavior
Length Scaled - Substrate Current
Length Scaled - Vth @ high Vd
Length Scaled - Output Resistance
Width Scaled - Vth
Narrow - DW
Small - D/S Resistance
Width Scaled - DW
Width Scaled - Bulk Charge Channel
Length Width Scaled - Vth @ low Vd
All Transistors - Finetuning Rds, DW, DL
Capacitance - Junction Bulk/Drain
Diode - Bulk/Drain
Temperature - All Parameters
Save Parameters
Extraction
Name
Length Scaled - Substrate Current
Transform
Length_Scaling/idvd/extract_all
Function Flow
E - ALPHA0, ALPHA1, BETA0
O - ALPHA0, ALPHA1, BETA0
Available Functions
E - ALPHA0, ALPHA1, BETA0
T - ALPHA0, ALPHA1, BETA0
O - ALPHA0, ALPHA1, BETA0

Extraction Strategy

This section describes two aspects of the extraction strategy: a group extraction and a physically oriented model parameter extraction.

Group Extraction Strategy

A major enhancement of the BSIM3v3 model compared to older simulation models is that one set of model parameters covers the whole range of channel lengths and channel widths of a certain process that can be used in circuit designs. Many effects in the BSIM3 model depend very strongly on device dimensions such as the channel length and width. This is considered in the determination of model parameters in the BSIM3v3 Modeling Package through the use of a group extraction strategy.

Figure 146 shows the principle procedure of model parameter extraction as it was used in older models like the MOS Level 3 model. The model parameter P_x is determined from the measured electrical behavior of one single test transistor. The measured data is transformed in such a way that P_x can be determined with regression methods.

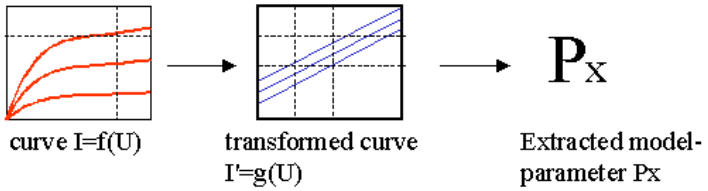


Figure 146 Model Parameter Extraction from Single Devices

In contrast, the group extraction strategy, which is shown in Figure 147, uses the measured electrical behavior of several test transistors with different gate lengths and gate widths.

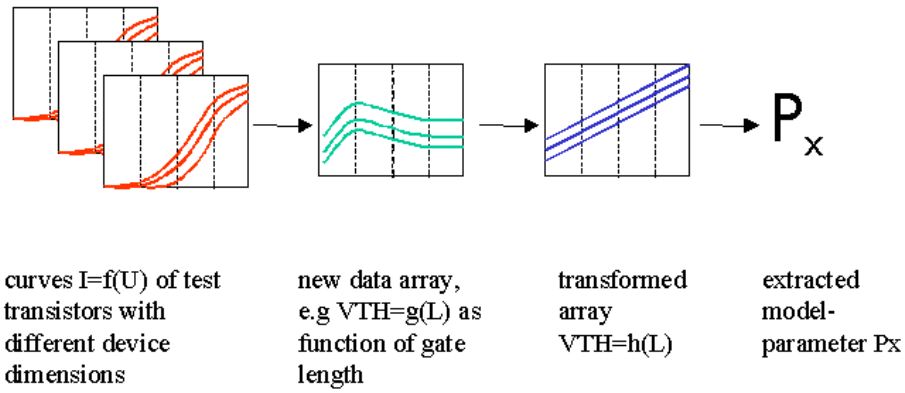

Px
curves I=f(U) of test transistors with different device dimensions
new data array, e.g VTH=g(L) as function of gate length
transformed array VTH=h(L)
extracted model-parameter Px

Figure 147 Group Extraction Strategy

In a first step, intermediate values like the threshold voltage V_{th} are determined and stored in a new data array as a function of gate length. In the next step, this new data array is transformed in such a way that the model parameters P_x can be determined with regression methods. Parameters extracted with this method describes the behavior very well of the devices in a wide range of channel lengths and channel widths.

Physically Oriented Model Parameter Extraction

For the determination of device model parameters from measured I-V or C-V curves, usually two general principles are applied—the optimization of the simulated device behavior or the parameter extraction based on the device equation.

The basis of the optimization process is the simulation of a device with exactly the same inputs (voltages, currents) that are used to measure the device. The error between simulated and measured data is the cost function for the optimization algorithms, which changes certain model parameters of the device, re-simulates it, and checks whether the error has increased or decreased. The advantage of this procedure is that the fitting between the measured curves and the simulated ones can be very good because the optimizer always tries to minimize this difference. However, in order to achieve this very good fitting, the optimization algorithm can give the model parameter physically unreasonable values. Another disadvantage is that many optimization algorithms are not able to find the global minimum of the failure function, which is the difference between measurement and simulation, and the success of the optimization depends on the start values of the model parameters. The last difficulty that can arise by using pure optimization algorithms for the model parameter determination is that the boundaries for the optimization process must be set very carefully. This means that the user of the optimization algorithm must have good knowledge about the device model and where the different model parameters have their influence on the device behavior in order to restrict the optimization process to a certain range of data.

In contrast to the optimization strategy, the extraction strategy is strictly based upon the device equations. If these device equations are physically oriented, as in the case of the BSIM3v3 model for MOS transistors, the extraction of the model parameters must give an accurate and realistic representation of the device physics. The basic idea of this extraction strategy is to transform measured data into such a form that model parameters of a certain part of the device

equations can be derived by mathematical regression methods. The extraction routines must therefore incorporate much more knowledge about the model and its behavior. Generally, model parameters extracted in this way are more realistic and physically oriented. However, the fitting between the measured and simulated curves can be less accurate than in the case of an optimization, because the extraction method gives a realistic physical representation of the device while the optimization only targets a minimum error between measurement and simulations.

Figure 149 shows the principle data flow of such an extraction routine for the short channel model parameters DVT0 and DVT1. In this example, the threshold voltage V_{th} of several test transistors with different gate lengths is determined and stored in an intermediate data array. The short channel effect ΔV_{th} is isolated in the next step from V_{th} as a function of gate length and bulk voltage V_{bs}. Figure 148 shows ΔV_{th} as a function of those two variables. Only a subset of this data array is used for the determination of DVT0 and DVT1, and the boundaries for defining this subset are set by the extraction routine. As shown in the flowchart in Figure 149, different results from this action are possible. In the first case, no data point is available for the extraction and the user is informed with a warning message. This may occur for instance after measurement errors or with old CMOS processes that do not show a short channel effect. As a further possible result, only one usable data point is returned. From this data point, one model parameter can be determined while the second one has to be set to its default value.

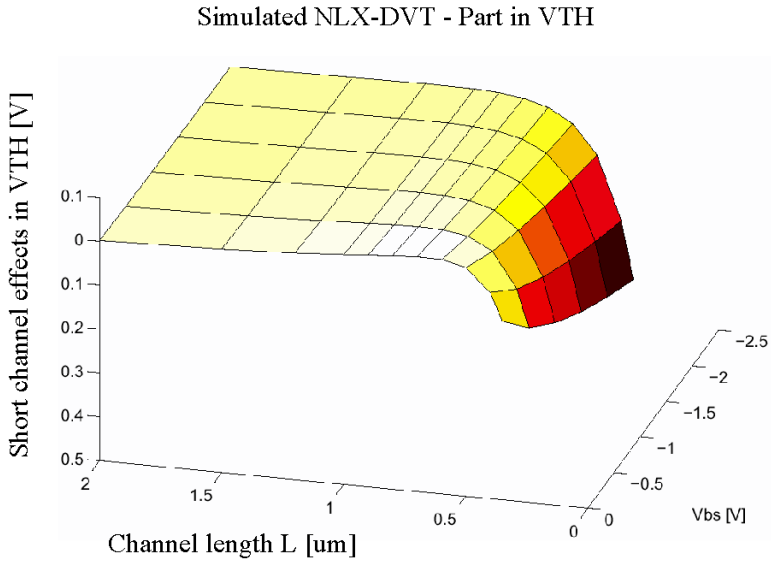


Figure 148 Short Channel Effects in V_{th} as a Function of Gate Length and V_{bs}

In the normal case, a group of usable data points can be identified and transformed in such a way that DVT0 and DVT1 can be extracted through linear regression methods.

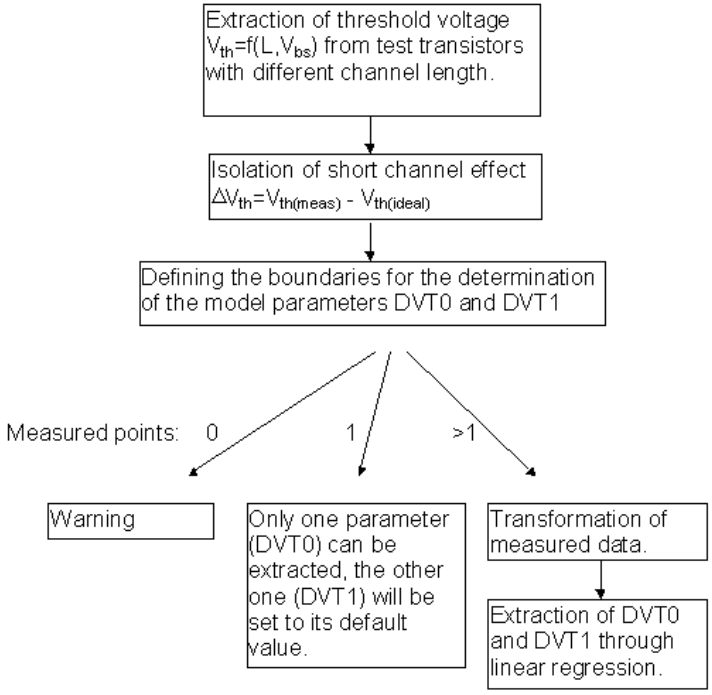

Extraction of threshold voltage $V_{th}=f(L,V_{bs})$ from test transistors with different channel length.
Isolation of short channel effect $\Delta V_{th}=V_{th(meas)} - V_{th(ideal)}$
Defining the boundaries for the determination of the model parameters DVT0 and DVT1
Measured points:
0
1
>1
Warning
Only one parameter (DVT0) can be extracted, the other one (DVT1) will be set to its default value.
Transformation of measured data.
Extraction of DVT0 and DVT1 through linear regression.

Figure 149 Extraction of Short Channel Effect Parameters DVT0, DVT1

Binning of Model Parameters

Usage of binned models in a simulator

The binning idea

The idea of binning is to provide different model parameter sets for a scalable model (e.g., a MOS device) according to the device dimensions. In the case of MOSFETs, the validity of such a parameter set is determined by LMIN, LMAX, WMIN, WMAX for each bin.

Major commercial simulators like HSPICE, Spectre, and ADS support the binning feature for semiconductor models. However, it is not included in standard UC Berkeley SPICE3f5!

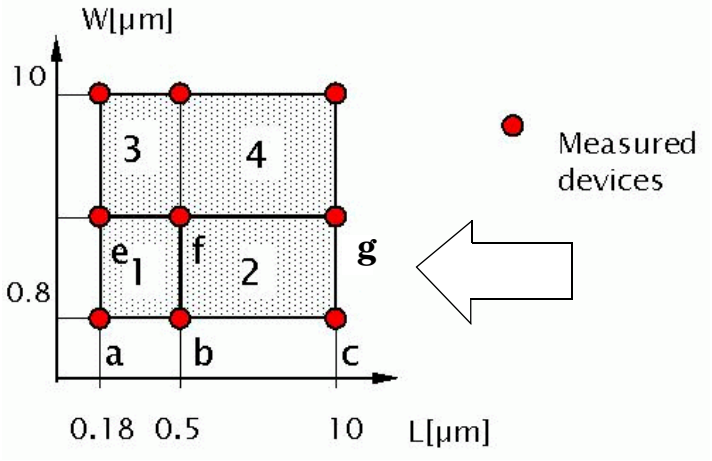


Figure 150 Binned model according to the measured devices

Lets take the example shown in the diagram above: we have 4 different bins with 4 different parameter sets. If we look only at the bins with the smallest width (see the arrow above), we still have 2 different parameter sets: set #1 and set #2.

The simulator would take the bins according to the following table:

Table 24 Bin Conditions

L < LMIN(bin1)	Error: not specified!
LMIN(bin1) ≤ L < LMAX(bin1)	BIN1 interpolated
LMIN(bin2) ≤ L < LMAX(bin2)	BIN2 interpolated
L ≥ LMAX(bin2)	Error: not specified!

Please note: LMAX(bin1) = LMIN(bin2)

The simulator now calculates an effective model parameter P(L) from the different binned parameter sets and the actual gate length of the device to simulate.

In addition, inside a certain bin, the parameter itself is interpolated so that we end up with the following diagram:

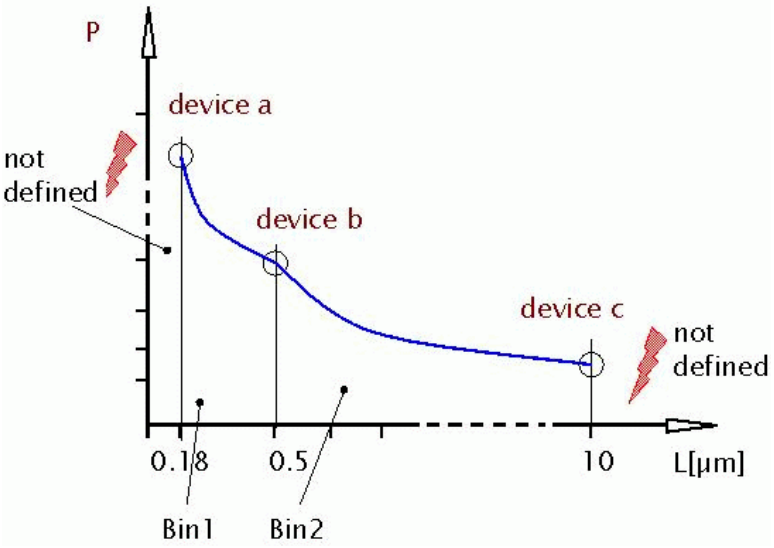


Figure 151 Calculation of binned model parameters

Advanced binning approaches

As the diagram in Figure 151 clearly shows, the model is defined only inside the gate length of the characterized devices. This is a critical condition, because the following two scenarios are very common for MOS devices:

- It is very usual to use a transistor with, for example, L = 10 µm as the largest measured device and to extrapolate the parameter set to devices with larger gate lengths. This is not a problem because the 10 um transistor already behaves like an ideal MOS transistor without short and narrow channel effects.
- For statistical simulations, the gate length and widths are overlaid by a statistical variation to reflect variations in lithography. If gate length or gate width are already at the boundary of the available model bins, this would not work.

Both described effects would cause no problem using the normal BSIM3 or BSIM4 model without binning. However, having the restrictions of the binning implementation in the simulators, the following two alternatives would help to overcome this bottleneck.

Extension of binning to include virtual devices

The first idea is to add additional model sets for areas in the L-W-space, which are not fully covered by measured devices. The following diagram shows such a scenario:

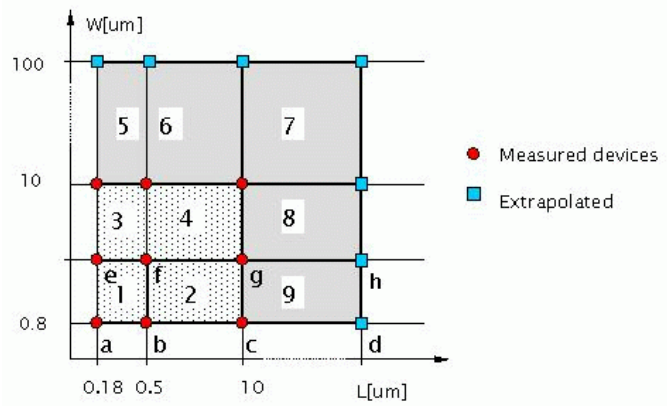


Figure 152 Extension of binning

The binned model parameter sets for region 1 through 4 have been determined from measured devices. Now, for the generation of the parameter set of region 9, it is assumed that the parameters for device ***d*** are equal to the parameters of device ***c*** and parameters from ***h*** are equal the parameters from ***g***. The gate length of ***d*** and ***h*** are selected so large, that they cover all useful applications. The diagram in Figure 153 shows the principal calculation of the parameter P over an available range of gate lengths L.

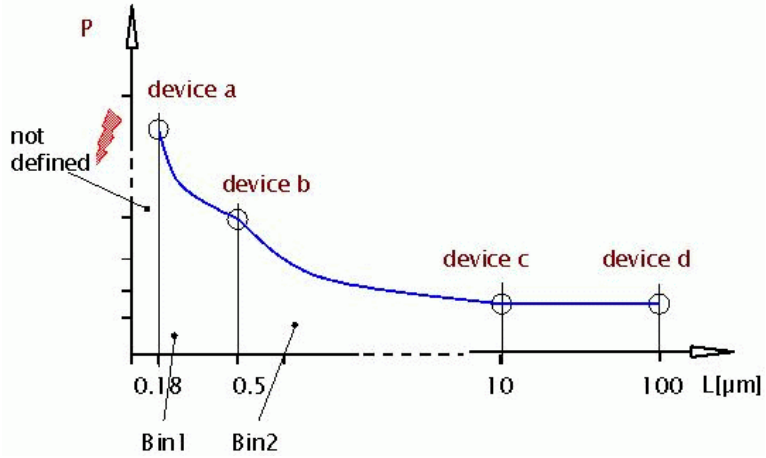


Figure 153 Calculation of binned model parameters with extensions

Extraction of binned model parameters

General algorithm

- For each device D_i, at the boundary of the bins, the original parameter $Po_i = P_i$ is determined using a circuit/model parameter set without the binning feature!
- For each bin, the parameter is interpolated for the actual length and width according to the following equation:

  $$P(L, W)_i = P_i + PL_i/L_{eff} + PW_i/W_{eff} + PP_i/(W_{eff}\ L_{eff})$$

- The binning parameters P_i, PL_i, PW_i, PP_i for one bin must be determined from the original parameters Po_i and Po_{i+1}. This is also very important to make sure that the parameter P is continuous at the boundary between two bins. That's the reason why the devices at the edges of a bin are used to determine the parameters!

Po_i	original model parameter, for example, VTH0 extracted separately for each device

$P(L)_i$	finally used interpolated model parameter inside a bin (internal SPICE value)
P_i	constant model parameter inside a bin (in model parameter set) Please note: this is the same parameter as Po_i but with a different meaning!
PL_i	length dependant model parameter inside a bin (in model parameter set)
PW_i	width dependant model parameter inside a bin (in model parameter set)
PP_i	length-width dependant model parameter inside a bin (in model parameter set)
L_{eff}	effective gate length
W_{eff}	effective gate width

For BSIM3:

$$L_{eff} = L_{des} - 2\left(LINT + \frac{LL}{L_{des}^{LLN}} + \frac{LW}{W_{des}^{LWN}} + \frac{LWL}{L_{des}^{LLN} W_{des}^{LWN}} \right)$$

$$W_{eff} = W_{des} - 2\left(WINT + \frac{WL}{L_{des}^{WLN}} + \frac{WW}{W_{des}^{WWN}} + \frac{WWL}{L_{des}^{WLN} W_{des}^{WWN}} \right)$$

For BSIM4:

$$L_{eff} = L_{des} + XL - 2\left(LINT + \frac{LL}{L_{des}^{LLN}} + \frac{LW}{\left(\frac{W_{des}}{NF}\right)^{LWN}} + \frac{LWL}{L_{des}^{LLN}\left(\frac{W_{des}}{NF}\right)^{LWN}} \right)$$

$$W_{eff} = \frac{W_{des}}{NF} + XW - 2\left(WINT + \frac{WL}{L_{des}^{WLN}} + \frac{WW}{\left(\frac{W_{des}}{NF}\right)^{WWN}} + \frac{WWL}{L_{des}^{WLN}\left(\frac{W_{des}}{NF}\right)^{WWN}} \right)$$

Implementation into the BSIM3 / 4 Modeling Packages

Output for the selected simulator

The output of the BSIM3/BSIM4 Modeling Packages is ready for use with a simulator. One of the major problems is that the basic SPICE3F5 simulator of UC Berkeley does not include the binning features of ADS, HSPICE, or Spectre. Therefore, binning will be limited to those commercial simulators!

The following listing shows a typical binned library for ADS:

```
; example ADS BinModel
; Min[ , ] (inclusive)
; Max[ , ] (exclusive, inclusive if Max=Min)
;
model my_nmos BinModel \
      Model[1]="my_nmos1" \
      Model[2]="my_nmos2" \
      Model[3]="my_nmos3" \
      Model[4]="my_nmos4" \
      Model[5]="my_nmos5" \
      Model[6]="my_nmos6" \
      Model[7]="my_nmos7" \
      Model[8]="my_nmos8" \
      Model[9]="my_nmos9" \
      Param[1]="Length" \
      Param[2]="Width" \
      Min[1,1]=L1 Max[1,1]=L2 Min[1,2]=W1 Max[1,2]=W2 \
      Min[2,1]=L2 Max[2,1]=L3 Min[2,2]=W1 Max[2,2]=W2 \
      Min[3,1]=L3 Max[3,1]=L4 Min[3,2]=W1 Max[3,2]=W2 \
      Min[4,1]=L1 Max[4,1]=L2 Min[4,2]=W2 Max[4,2]=W3 \
      Min[5,1]=L2 Max[5,1]=L3 Min[5,2]=W2 Max[4,2]=W3 \
      Min[6,1]=L3 Max[6,1]=L4 Min[6,2]=W2 Max[4,2]=W3 \
      Min[7,1]=L1 Max[7,1]=L2 Min[7,2]=W3 Max[4,2]=W4 \
      Min[8,1]=L2 Max[8,1]=L3 Min[8,2]=W3 Max[4,2]=W4 \
      Min[9,1]=L3 Max[9,1]=L4 Min[9,2]=W3 Max[4,2]=W4
;
model my_nmos1 MOSFET NMOS=1 PMOS=0 etc ...
model my_nmos2 MOSFET NMOS=1 PMOS=0 etc ...
model my_nmos3 MOSFET NMOS=1 PMOS=0 etc ...
model my_nmos4 MOSFET NMOS=1 PMOS=0 etc ...
model my_nmos5 MOSFET NMOS=1 PMOS=0 etc ...
model my_nmos6 MOSFET NMOS=1 PMOS=0 etc ...
model my_nmos7 MOSFET NMOS=1 PMOS=0 etc ...
model my_nmos8 MOSFET NMOS=1 PMOS=0 etc ...
model my_nmos9 MOSFET NMOS=1 PMOS=0 etc ...
```

Definition of binning areas

One of the major disadvantages of the binning approach is, that the scalability feature of a model is not fully taken into account. With the binning approach, all binned parameters are interpolated using the same functions 1/L and 1/W.

Typically a scalable model behavior, for example, the threshold voltage, is replaced by the binning approach. The following example will make this more clear:

$$V_{th} = VTH0 + \left(K_{1ox}\cdot\sqrt{\Phi_s - V_{bseff}} - K1\cdot\sqrt{\Phi_s}\right)\sqrt{1+\frac{LPEB}{L_{eff}}} - K_{2ox}V_{bseff}$$
$$+ K_{1ox}\left(\sqrt{1+\frac{LPE0}{L_{eff}}} - 1\right)\sqrt{\Phi_s} + \left(K3 + K3B\cdot V_{bseff}\right)\frac{TOXE}{W_{eff}' + W0}\Phi_s$$
$$- 0.5\cdot\left[\frac{DVT0W}{\cosh\left(DVT1W\frac{L_{eff}W_{eff}'}{l_{tw}}\right)-1}\right]\left[\frac{DVT0}{\cosh\left(DVT1\frac{L_{eff}}{l_t}\right)-1}\right](V_{bi} - \Phi_s)$$
$$- \frac{0.5}{\cosh\left(DSUB\frac{L_{eff}}{l_t}\right)-1}\left(ETA0 + ETAB\cdot V_{bseff}\right)\cdot V_{ds}$$
$$- nv_t\cdot\ln\left(\frac{L_{eff}}{L_{eff} + DVTP0\cdot\left(1+e^{-DVTP1\cdot V_{ds}}\right)}\right)$$

The threshold voltage of BSIM4 is given above. It is a complex equation that describes length and width related effects. In a binned model, parameters describing those effects (DVT0, DVT1, LPE0, e.t.c.) are normally not used. Instead the basic threshold voltage parameters VTH0, K1, and K2 together with the binning extensions PVTH0, LVTH0, WVTH0, ..., WK2 are describing these effects.

If we have a Vth-function like in the following figure, it is clear that a proper selection of binning areas is necessary to cover this behavior!

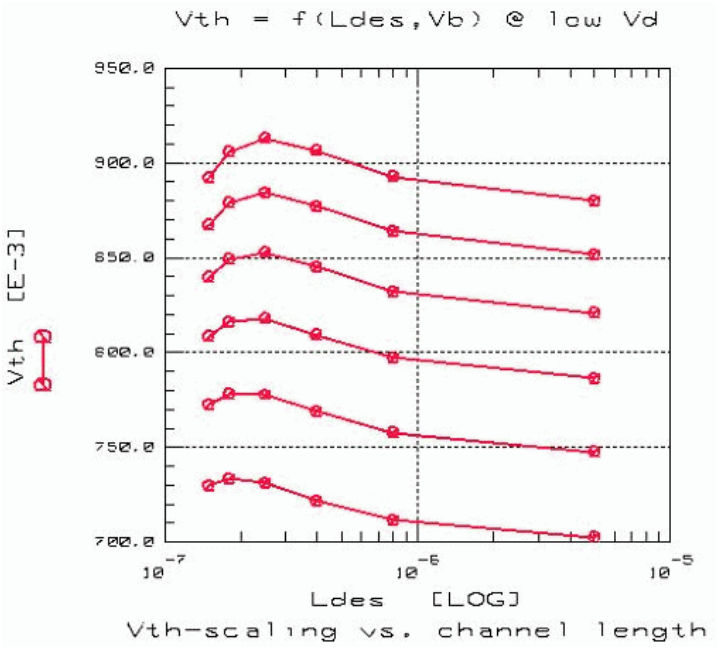


Figure 154 Typical V_{th} behavior of a 0.18um CMOS process (n-type)

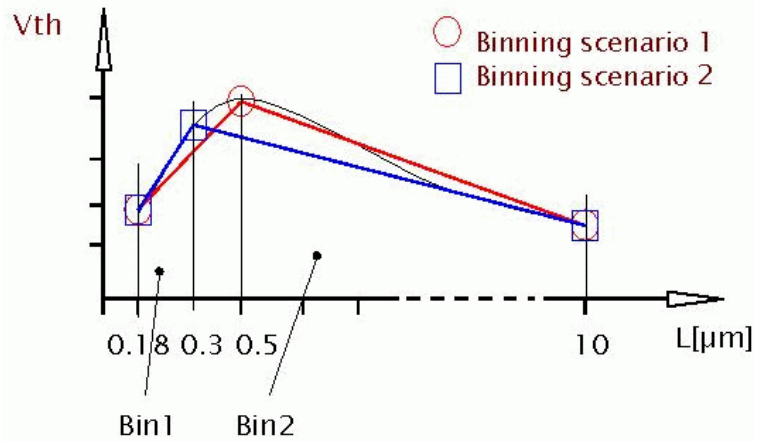


Figure 155 Two different binning scenarios

The previous diagram clearly shows the difficulty in defining proper boundaries for the different binning areas. While the binning scenario 1 covers the typical behavior of Vth, the second scenario would miss the point of maximum Vth.

To verify the correct behavior, additional devices between the binning boundaries are necessary, especially in the critical areas with minimum gate lengths and widths as outlined below:

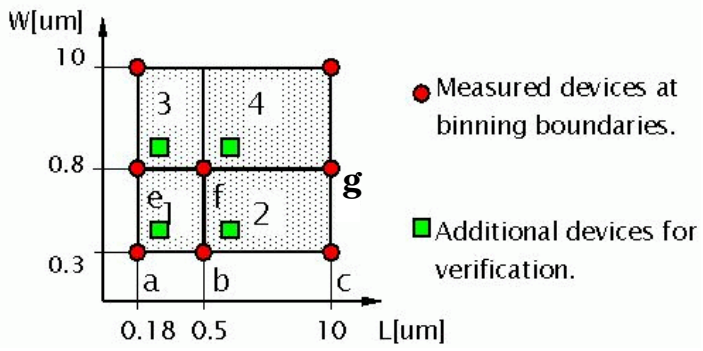


Figure 156 Devices for verification

Importing older version BSIM3v3 Files

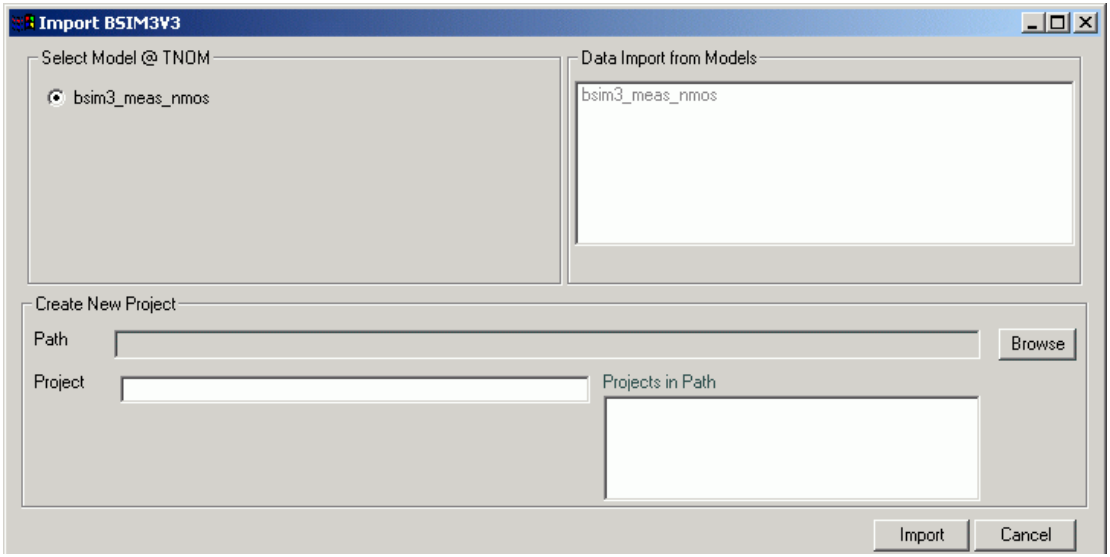
This section is intended for users wishing to import model files created with former versions of the BSIM3 Modeling Package (the non-graphic version) into the new BSIM3 Modeling Package using the GUI.

If you have a model file created with a former, non-graphic version of the BSIM3 Modeling Package, please proceed as described in the following example.

Open *../examples/model_files/mosfet/BSIM3/BSIM3_DC_CV_Measure* as well as the BSIM3v3-Model-File.mdl you wish to import.

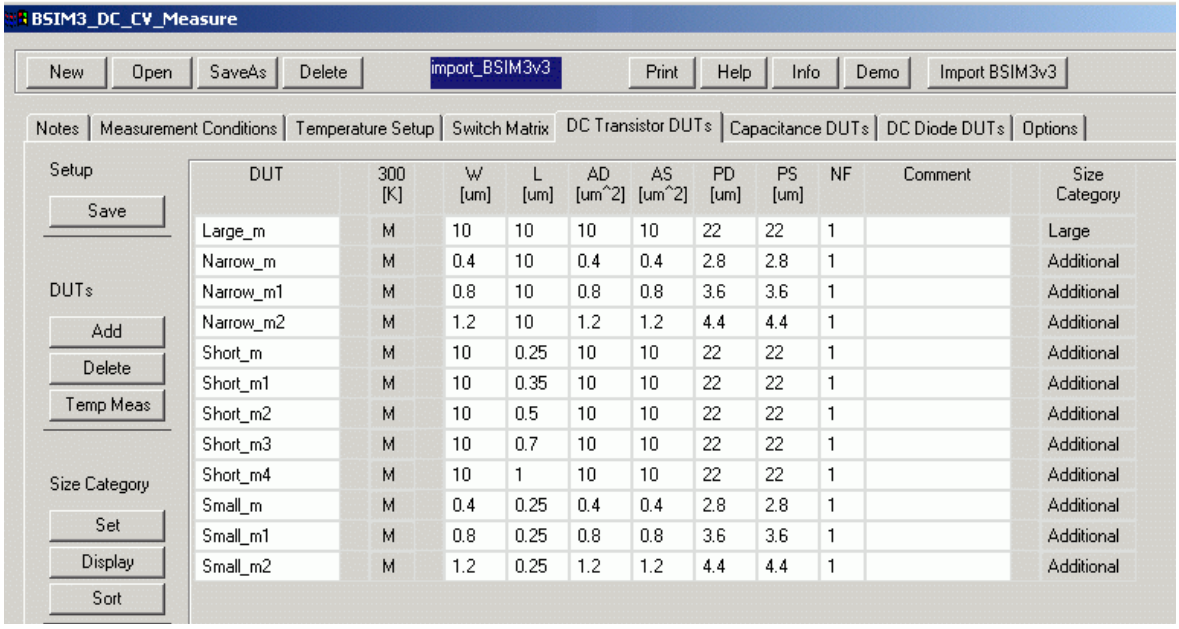
For example, if the BSIM3v3.mdl file is located in *.../examples/model_files/mosfet/bsim3v3/examples/dc_modeling/ MASTER_MEAS_nmos.mdl*, open that one.

The next step is to choose *ImportBSIM3v3* in the header of the new GUI BSIM3_DC_CV_Measure.mdl. You will get a prompt as shown in the following figure:



Enter the name and location of the new BSIM3 project to be created and choose *Import*.

After the data is imported, you will get a message stating that the BSIM3v3-data has been successfully imported and you should go to the DC Transistor DUTs-folder, which looks like Figure 157.



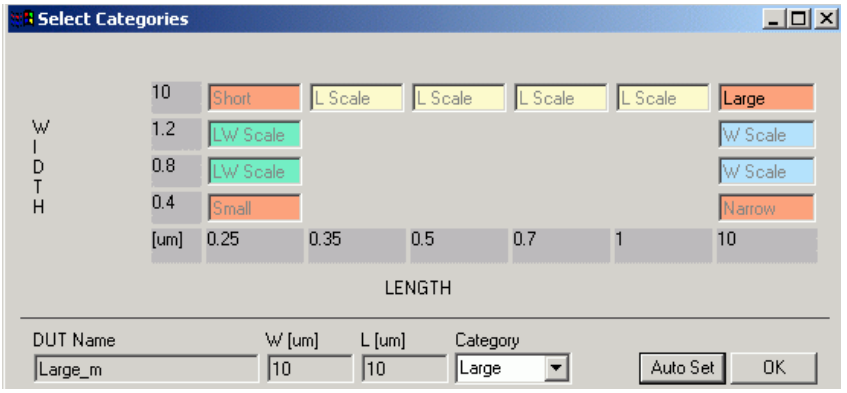
DUT	300 [K]	W [um]	L [um]	AD [um^2]	AS [um^2]	PD [um]	PS [um]	NF	Comment	Size Category
Large_m	M	10	10	10	10	22	22	1		Large
Narrow_m	M	0.4	10	0.4	0.4	2.8	2.8	1		Additional
Narrow_m1	M	0.8	10	0.8	0.8	3.6	3.6	1		Additional
Narrow_m2	M	1.2	10	1.2	1.2	4.4	4.4	1		Additional
Short_m	M	10	0.25	10	10	22	22	1		Additional
Short_m1	M	10	0.35	10	10	22	22	1		Additional
Short_m2	M	10	0.5	10	10	22	22	1		Additional
Short_m3	M	10	0.7	10	10	22	22	1		Additional
Short_m4	M	10	1	10	10	22	22	1		Additional
Small_m	M	0.4	0.25	0.4	0.4	2.8	2.8	1		Additional
Small_m1	M	0.8	0.25	0.8	0.8	3.6	3.6	1		Additional
Small_m2	M	1.2	0.25	1.2	1.2	4.4	4.4	1		Additional

Figure 157 Imported MASTER_MEAS_nmos.mdl from a former BSIM3v3 project

All transistors with the exception of the *Large* one are set to *Additional*.

NOTE The DUT names already contain the *functionality*! This may be confusing, but you can change the names after the import procedure.

When starting the *Size Category* **Set** action, the following GUI appears:



Perform *Auto Set* and select *Large_m* as the Large Transistor inside the *Set Category* window.

Now the size categories are defined automatically and the resulting DCTransistorDUTs folder is shown in Figure 158.

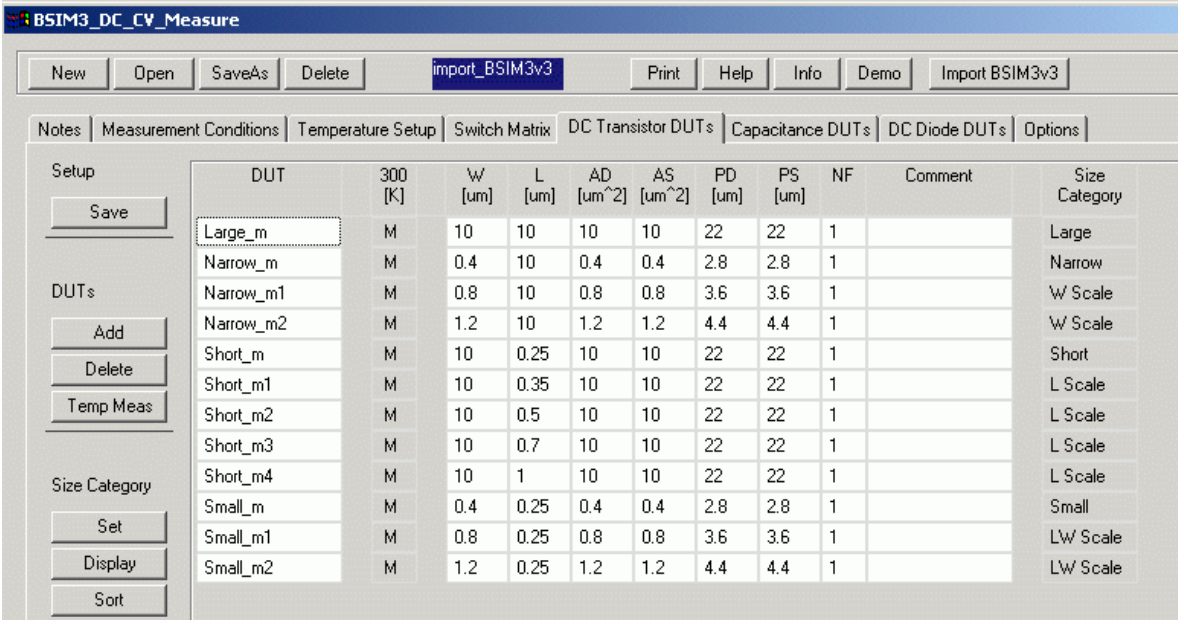


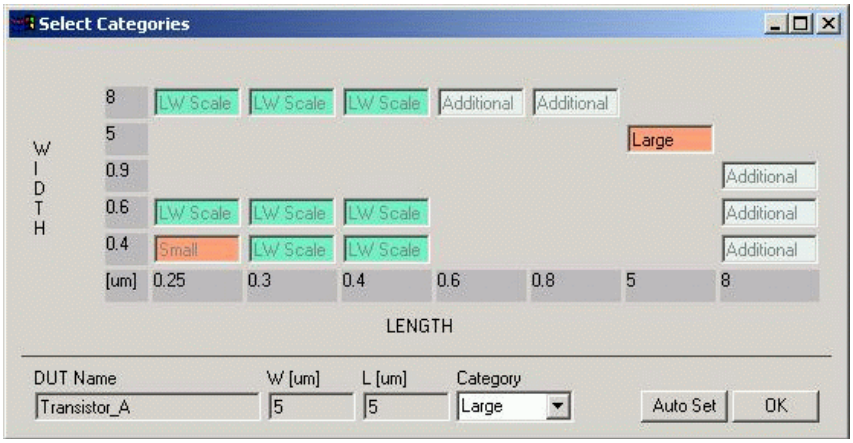
Figure 158 DC Transistor DUTs-folder after successful set size categories

You can now save the project and proceed as you would with measured project data.

Hints

If you do not have such clearly defined test structures at hand, you may have to select the size category manually. The following example demonstrates this.

In the following figure, the *Large* transistor does not fit the max. L of the *Length scaled* and the max. W of the *Width scaled* devices because it's size is 5 µm * 5 µm. Therefore, the automatic assignment of the size categories does not lead to satisfying results.



To set the size manually, click at a certain device and change the category by selecting the appropriate category from a pull-down menu in the middle of the bottom row of buttons in the *Select Categories* window.

References

1 "BSIM3 Manual," University of California at Berkeley, December 2001

2 "Characterization System for Submicron CMOS Technologies," JESSI Reports AC41 94-1 through 94-6

3 Peter Klein, “A consistent parameter extraction method for deep submicron MOSFETs,” Proc. 27th European Solid-State Device Research Conference, Stuttgart, Germany, 1997

4 "Layout Rules for GHz Probing", Application Note Cascade Microtech

5 F. Sischka, "Deembedding Toolkit," Agilent, GmbH, Böblingen, Germany

6 File: "deemb_short_open.mdl" in IC-CAP examples, Agilent EEsof

7 W. Liu et al., “R.F. MOSFET Modeling Accounting for Distributed Substrate and Channel Resistances with Emphasis on the BSIM3v3 SPICE Model,” Proc. IEEE IEDM, 1997

8 C. Enz, “MOS Transistor Modeling for RF IC Design”, Silicon RF IC: Modeling and Simulation Workshop, Lausanne, Switzerland, 2000

9 M.Jamal Deen (Ed.), T.A.Fjeldly, “CMOS RF Modeling, Characterization and Applications”, Worldscientific, Co-authors: F.Sischka and T.Gneiting

10 W. Liu, “MOSFET Models for SPICE Simulation, including BSIM3v3 and BSIM4”, Wiley-Interscience, 2001

How to get the BSIM3v3 manual from University of Berkeley/California:

University of Berkeley/California provides an easy way to get a free copy of the BSIM3v3 manual and the BSIM3v3 source code from their world-wide web home page:

http://www-device.EECS.Berkeley.EDU/~bsim3/

Other useful internet addresses:

Advanced Modeling Solutions:

http://www.admos.de

Agilent EEsof homepage

http://www.agilent.com/find/eesof/

Copyright

BSIM3 has been developed by the Device Research Group of the Department of Electrical Engineering and Computer Science, University of California, Berkeley and copyrighted by the University of California.

5 BSIM4 Characterization

Basic Effects Modeled in BSIM4 341
Key Features of the BSIM4 Modeling Package 342
DC Behavioral Modeling 343
CV Modeling 376
Structure of the BSIM4 RF Simulation Model 383
398
Tutorials for the BSIM4 Modeling Package 400
SPICE Model Parameters 406
References 425

This chapter provides a theoretical background for the BSIM4 model. It is based on the model revision BSIM4.3.0, released by the University of California at Berkeley on May 9, 2003. Using the Modeling Packages is described in Chapter 3, “Using the BSIM Modeling Packages.”

BSIM4.3.0 is developed to address several new issues in modeling sub-0.13 micron CMOS technology and RF high-speed CMOS circuit simulation.

In addition to several bug fixes, BSIM4.3.0 has the following major improvements and additions over BSIM4.2.1:

- Shallow trench isolation has been adopted by a new scalable stress effect model for process induced stress effects. Performance thus becomes a function of the active area geometry and the location of the device inside the active area.

- Velocity saturation, velocity overshoot and source end velocity limit effects are taken into account through a unified current-saturation model.
- The use of a new temperature model allows simulation of temperature effects on saturation velocity, mobility, and S/D resistances.
- The holistic thermal noise model has been enhanced.
- The forward body bias model has been improved.
- The gate direct tunneling model has been extended to multiple-layer gate dielectrics.

The complete list of bugs and fixes and the users who reported them, the BSIM4.3.0 source code, BSIM4.3.0 user manual, BSIM4.3.0 new enhancement document and testing examples can be downloaded at [1]:

http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html

What's new inside the BSIM4 Modeling Package:

This section lists the enhancements and changes made to the Modeling Package for each revision since IC-CAP 2002. They are listed in reverse order so that the new version is on top, followed by former versions.

New features in the BSIM4 Modeling Package, Rev. IC-CAP 2004, spring 2005

NOTE

The supported model is now BSIM4.3.0, released by UCB in May 2003.

1.) General

The new effects modeled in BSIM4.3.0 are included as well as some improvements in handling the Modeling Package.

- Shallow Trench Isolation (STI) effects
- New temperature model
- Enhancements in the Holistic noise modeling
- Multilayer Gate tunneling current
- Tolerances in Binning
- Scalable Model has been enhanced for specific layouts using so called "Horseshoe contacts"
- Consistency check of DC measurement data included
- The time to loading a new project has been dramatically reduced
- Extractions for BSIM4.3.0 specific parameters have been included
- New scheme to define de-embedding structures

New features in the BSIM4 Modeling Package, Rev. IC-CAP 2004, January 2004

1.) General

The BSIM4 Modeling Package can now generate model cards and scalable RF models for the following simulators:

Spice3 (delivered with IC-CAP)

Advanced Design System

Hspice

Spectre

All temperatures in setups and documentation are now given in [K] instead of [degree C]. Many suggestions from beta and first time users have been included into the Modeling Package.

2.) BSIM4_DC_CV_Measure

The Keithley switching matrix models K707 and K708a are supported.

The maximum compliance values can be defined together with the other measurement settings.

Three new functions are implemented to drive the *BSIM4_DC_CV_Measure* module from a wafer prober control macro. An example for such a control macro can be found in *'.../examples/model_files/mosfet/bsim4/examples/waferprober/prober_control.mdl'*

3.) BSIM4_DC_CV_Extract

A complete new extraction flow is implemented. A certain extraction group (e.g., 'Basic VTH, Mobility') can be invoked several times with different configurations.

4) BSIM4_RF_Extract

A complete new extraction flow is implemented. Please see the above noted *BSIM4_DC_CV_Extract* for more details.

The automatic generation of HTML files has been enhanced to include a navigation tree through all results.

5) BSIM4_DC_CV_Tutorial

The model file *BSIM4_DC_CV_Tutorial* allows the user to evaluate all physical effects of the BSIM4 model. It provides links to the documentation to invoke the appropriate description of a certain effect while visualizing it using the tuner feature.

The model file can be found under

../mosfet/bsim4/tutorial/BSIM4_DC_CV_Tutorial.mdl

New features in the BSIM4 Modeling Package, Rev. IC-CAP 2002, March 2003

The major enhancement in this update is the possibility to use the same set of measured data to extract parameters for the BSIM3 Model as well as for the BSIM4 Model.

This means, data generated by the *BSIM3_DC_CV_Measure* module can be imported into the *BSIM4_DC_CV_Extract* module and vice versa.

This feature allows the user to extract model parameters for use with the BSIM3 model as well as the BSIM4 model according to his needs, using only one set of measured data.

Moreover, the flow inside an extraction group can be specified in any desired order to get highest flexibility in adopting a specific parameter extraction to a certain process.

The automatic generation of binned model files is now supported. A new folder '*Binning*' in the '*BSIM4_DC_CV_Extract*' module allows the specification of binning areas as well as extended binning. Final circuits are generated for simulators supporting binned model parameters: Hspice, Spectre and ADS.

The automatic generation of HTML files has been enhanced to include a navigation tree through all results. In addition, all measured data at each temperature for each device is compared with the simulated results.

The new IC-CAP feature "Plot Optimizer" is supported through the use of an additional folder *Plot Optimizer* which allows the entering of devices and setups for a final fine tuning approach.

A new function is implemented to extract multiple projects in a batch mode. This can be very useful for statistical modeling, where a large number of model parameter sets have to be generated for the same type of devices but from different measured test chips. Please see the macro 'Example_Wafer_Extraction' in the *BSIM4_DC_CV_Extract.mdl* file.

Parameter extractions have been steadily enhanced due to user's feedback.

Basic Effects Modeled in BSIM4

- Short and narrow channel effects on threshold voltage
- Non-uniform doping effects
- Mobility reduction due to vertical field
- Bulk charge effect
- Carrier velocity saturation
- Drain induced barrier lowering (DIBL)
- Channel length modulation (CLM)
- Substrate current induced body effect (SCBE)
- Parasitic resistance effects
- Quantum mechanic charge thickness model

Enhanced drain current model

- VTH model for pocket/retrograde technologies
- New predictive mobility model
- Gate induced drain leakage (GIDL)
- Internal/external bias-dependent drain source resistance

RF and high-speed model

- Intrinsic input resistance (Rgate) model
- Non-Quasi-Static (NQS) model
- Holistic and noise-partition thermal noise model
- Substrate resistance network
- Calculation of layout-dependent parasitic elements
- Asymmetrical source/drain junction diode model
- I-V and breakdown model
- Gate dielectric tunneling current model

Key Features of the BSIM4 Modeling Package

- The new graphical user interface in Agilent's IC-CAP enables the quick setup of tests and measurements followed by automatic parameter extraction routines.
- A new data management concept allows a powerful and flexible handling of measurement data using an open and easy data base concept.
- The powerful extraction procedures can be easily adapted to different CMOS processes. They support all possible configurations of the BSIM4 model.
- Quality assurance procedures are checking every step in the modeling flow from measurements to the final export of the SPICE model parameter set.
- The fully automatic generation of HTML reports is included to enable web publishing of a modeling project.
- The modeling package supports SPICE3e2 and major commercial simulator formats such as HSPICE, Spectre, or Agilent's ADS.

The Modeling Package Supports Measurements on

- Single finger (normal) transistors
- Parasitic diodes
- Capacitances:
  - Oxide and Overlap
  - Bulk-Drain and Source-Drain junction
  - Intrinsic
- RF multifinger transistors

The Modeling Package Supports Extractions for

- Basic transistor behavior
- Parasitic diodes
- Capacitances
- RF behavior (S-parameters)

DC Behavioral Modeling

This section provides a theoretical background of the BSIM4 DC model. You will find some basic device equations together with some explanations on model selectors used inside the BSIM4 model and the BSIM4 Modeling Package.

At the end of this section you'll find a table listing all of the model parameters added in BSIM4.3.0 together with their default values as well as a table containing the parameter set used in version BSIM4.2.1.

Since this theoretical section can only be of introductional character, we strongly recommend that you consult the manual from the University of California, Berkeley for a detailed description of device equations and relevant parameters [1] as well as further literature (see "References" on page 425).

Threshold Voltage Model

The complete threshold voltage model equation implemented in the BSIM4 model for SPICE is:

$$V_{th} = VTH0 + K1 \cdot [Part1] - K2 \cdot [Part2] \quad \textbf{(88)}$$
$$+ (K3 + K3B \cdot V_{bseff}) \cdot \frac{TOXE}{W'_{eff} + W0} \cdot \Phi_S$$
$$- \left(\frac{1}{2} \cdot Part3 \cdot (V_{bi} - \Phi_s)\right) - \left(\frac{1}{2} \cdot Part4\right)$$

The equation above contains some shortcuts for better readability ($[Part1, Part2, Part3 and Part4]$). Expanded they read:

$$Part1 = K1 \cdot \sqrt{1 + \frac{LPEB}{L_{eff}}} \cdot \left(\frac{TOXE}{TOXM} \cdot \sqrt{\Phi_S - V_{bseff}} - \sqrt{\Phi_S}\right)$$
$$+ \left(\frac{TOXE}{TOXM} \cdot \sqrt{\Phi_S} \cdot \left(\sqrt{1 + \frac{LPE0}{L_{eff}}} - 1\right)\right)$$

$$Part2 = \frac{TOXE}{TOXM} \cdot V_{bseff}$$

$$Part3 = \left[\frac{DVT0W}{\cosh\left(DVT1W \cdot \frac{L_{eff}W_{eff}}{l_{tw}}\right) - 1} + \frac{DVT0}{\cosh\left(DVT1 \cdot \frac{L_{eff}}{l_t}\right) - 1} \right]$$

$$Part4 = \frac{V_{ds}}{\cosh\left(DSUB \cdot \frac{L_{eff}}{l_{t0}}\right) - 1} \cdot (ETA0 + ETAB \cdot V_{bseff})$$

To set an upper boundary for body bias during simulations, the effective body bias has been introduced:

$$V_{bseff} = V_{bc} + \frac{1}{2}(V_{bs} - V_{bc} - \delta_1) + \frac{1}{2} \cdot (\sqrt{(V_{bs} - V_{bc} - \delta_1)^2 - 4\delta_1 V_{bc}}) \quad \textbf{(89)}$$

where $\delta_1 = 10^{-3} V$ and V_{bc} is the maximum allowable V_{bs} and is calculated from $(dV_{th})/(dV_{bs}) = 0$ to be

$$V_{bc} = \frac{9}{10} \cdot \left(\Phi_s - \frac{K1^2}{4 \cdot K2^2} \right)$$

Furthermore, there are some shortcuts used to make Equation 88 more readable.

built-in voltage of the Source/Drain regions:

$$V_{bi} = \frac{k_B \cdot T}{q} \cdot \ln\left(\frac{NDEP \cdot NSD}{n_i^2}\right)$$

characteristic length l_t:

$$l_t = \sqrt{\frac{\varepsilon_{si} \cdot TOXE \cdot \sqrt{\frac{2 \cdot \varepsilon_{si} \cdot (\Phi_s - V_{bs})}{q \cdot NDEP}}}{EPSROX}} \cdot (1 + DVT2 \cdot V_{bs})$$

and

$$l_{t0} = \sqrt{\frac{\varepsilon_{si} \cdot TOXE \cdot \sqrt{\frac{2 \cdot \varepsilon_{si} \cdot \Phi_s}{q \cdot NDEP}}}{EPSROX}}$$

$$l_{tw} = \sqrt{\frac{\varepsilon_{si} \cdot TOXE \cdot \sqrt{\frac{2 \cdot \varepsilon_{si} \cdot \Phi_s}{q \cdot NDEP}}}{EPSROX}} \cdot (1 + DVT2W \cdot V_{bs})$$

effective channel length and width, L_{eff} and W_{eff}:

$$L_{eff} = L_{drawn} - 2dL$$

$$W_{eff} = \frac{W_{drawn}}{NF} - 2dW$$

Table 25 Variables and parameters used in modeling threshold voltage

Equation Variable	BSIM4 Parameter	Description	Default Value
VTH0	VTH0	long channel threshold voltage at V_{bs} = 0	NMOS: 0.7 V PMOS: -0.7 V
K1	K1	first-order body effect coefficient	0.5 $\sqrt{V}$
K2	K2	second-order body effect coefficient	0
K3	K3	narrow width coefficient	80.0
K3B	K3B	Body effect coefficient of K3	0.0 1/ V
W0	W0	narrow width parameter	2.5E-6 m
LPE0	LPE0	lateral non-uniform doping parameter at V_{bs} = 0	1.74e-7
LPEB	LPEB	lateral non-uniform doping effect on K1	0 V

Table 25 Variables and parameters used in modeling threshold voltage (continued)

Equation Variable	BSIM4 Parameter	Description	Default Value
k_B		Boltzmann's constant ($k_B = 1,3807\times10^{-23}\frac{J}{K}$)	
T	T	absolute temperature in Kelvin	300
q		charge of an Electron ($q = 1,602\times10^{-19}C$	
L_{drawn}	-	channel length as drawn on mask	-
W_{drawn}	-	channel width as drawn on mask	-
NF	NF	number of gate fingers	1
TOXE	TOXE	electrical gate equivalent oxide thickness	3E-9m
TOXM	TOXM	Gate oxide thickness at which parameters are extracted	TOXE
DVT0	DVT0	first coefficient of short-channel effect on VTH	2.2
DVT0W	DVT0W	first coefficient of narrow-width effect on VTH for small channel length	0
DVT1	DVT1	second coefficient of short-channel effect on VTH	0.53
DVT1W	DVT1W	second coefficient of narrow-width effect on VTH for small channel length	5.3E6m
DVT2	DVT2	body-bias coefficient of short-channel effect on VTH	-0.032 1/V
DVT2W	DVT2W	body-bias coefficient of narrow-width effect on VTH for small channel length	-0.032 1/V
DSUB	DSUB	DIBL coefficient exponent in subthreshold region	DROUT
ETA0	ETA0	DIBL coefficient in the subthreshold region	0.08
ETAB	ETAB	body-bias for the subthreshold DIBL effect	-0.07 1/V
NDEP	NDEP	channel doping concentration at X_{dep0}, the depletion edge at $V_{bs} = 0$	1E17 cm^{-3}
NSD	NSD	doping concentration of the S/D diffusions	1e20 cm^{-3}
ε_{si}		relative dielectric constant of silicon	11.8

Table 25 Variables and parameters used in modeling threshold voltage (continued)

Equation Variable	BSIM4 Parameter	Description	Default Value
EPSROX	EPSROX	gate isolators relative dielectric constant (silicon dioxide)	3.9
Φ_S		surface potential	-
V_{bs}		Bulk-Source voltage	-
V_{ds}		Drain-Source voltage	-

The following sections provide equations for effects modeled in the complete equation above, Equation 88. Starting from the basic equation for long and wide channels, the effects of shrinking dimensions and substrate doping variations are modeled step by step.

Basic Threshold voltage equation

For long and wide channels the following equation is valid:

$$V_{th} = V_{FB} + \Phi_S + \gamma \cdot \sqrt{\Phi_S - V_{bs}} = \quad \textbf{(90)}$$

$$VTH0 + \frac{\sqrt{2q\varepsilon_{Si}N_{substrate}}}{C_{oxe}}\left(\sqrt{\Phi_S - V_{bs}} - \sqrt{\Phi_S}\right)$$

Equation 90 is valid under the following assumptions:

- constant substrate (channel) doping
- long and wide channel

Model parameters used for the equation above are listed in the table below:

Table 26 BSIM4 model parameters used in the basic threshold voltage equation

Equation Variable	BSIM4 Parameter	Description	Default Value
γ	GAMMA	body bias coefficient	
$N_{substrate}$	NSUB	uniform substrate doping concentration	6E16 cm^{-3}
C_{oxe}			
V_{FB}	VFB	flatband voltage	-1.0 V

If the substrate doping is not constant or if the channel is short and/or narrow, the basic equation should be modified. The following sections show modifications to the basic equation for non-uniform doping concentration and for short or narrow channel effects.

Non-Uniform Substrate Doping

If the substrate doping concentration is not uniform in vertical direction, the body bias coefficient γ is a function of the substrate bias and the depth from the interface. The threshold voltage in case of non-uniform vertical doping is:

$$V_{th} = VTH0 + K1(\sqrt{\varphi_S - V_{bs}} - \sqrt{\varphi_S}) - K2 \cdot V_{bs} \quad \textbf{(91)}$$

where $\varphi_S = \frac{4}{10} \cdot \frac{k_B T}{q} \cdot \ln\left(\frac{NDEP}{n_i}\right) + PHIN$

The doping profile is assumed to be a steep retrograde and is approximated in BSIM4. For details on how it is modeled, see the manual from UC Berkeley [1] starting on page 2-2. The model parameters K1 and K2 can be calculated from NSUB, XT, VBX, VBM, and so on. This is done the same way as in BSIM3v3. Details can be found on page 2-4 of the BSIM4 manual.

Table 27 Non-uniform substrate doping model parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
n_i		intrinsic carrier concentration in the channel region	
PHIN	PHIN	Non-uniform vertical doping effect on surface potential	0.0

Non-Uniform Lateral Doping: Pocket or Halo Implant

The doping concentration in this case varies from the middle of the channel towards the source/drain junctions. Shorter channel lengths will result in a roll-up of V_{th} from the rise of the effective channel doping concentration and the changes of the body bias effect. Those effects are considered using the following formulation:

$$V_{th} = VTH0 + K1 \cdot \left(\sqrt{\Phi_s - V_{bs}} - \sqrt{\Phi_s}\right) \cdot \sqrt{1 + \frac{LPEB}{L_{eff}}} - K2 \cdot V_{bs} + K1 \cdot \left(\sqrt{1 + \frac{LPE0}{L_{eff}}} - 1\right) \cdot \sqrt{\Phi_s} \quad \textbf{(92)}$$

Additionally, drain-induced threshold shift (DITS) has to be considered in long-channel devices using pocket implant. For V_{ds} in a range of interest, a simplified threshold voltage shift caused by DITS was implemented:

$$\Delta V_{th}(DITS) = -n\nu_t \cdot \ln\left(\frac{L_{eff}}{L_{eff} + DVTP0 \cdot \left(1 + e^{-DVTP1 \cdot V_{ds}}\right)}\right) \quad \textbf{(93)}$$

Short-Channel and Drain-Induced-Barrier-Lowering (DIBL) Effects

For shorter channels, the threshold voltage is more sensitive to drain bias (DIBL effect) and less sensitive to body bias because of reduced control of the depletion region.

The short channel effect coefficient is given by

$$\theta_{th}(L_{eff}) = \frac{1}{2\left(\cosh\left(\frac{L_{eff}}{l_t}\right) - 1\right)}$$

In BSIM3v3 this equation is approximated, which results in a phantom second roll-up when L_{eff} becomes very small. To avoid this effect, the exact formulation is used in BSIM4. Model flexibility is increased for different technologies with additional parameters introduced and the short-channel and drain-induced-barrier-lowering effects are modeled separately.

This leads to a short-channel effect coefficient of the form:

$$\theta_{th}(SCE) = \frac{DVT0}{2\left(\cosh\left(DVT1 \cdot \frac{L_{eff}}{l_t}\right) - 1\right)}$$

and a variation of V_{th} due to the short-channel effect of:

$$\Delta V_{th}(SCE) = -\left(\frac{DVT0}{2\left(\cosh\left(DVT1 \cdot \frac{L_{eff}}{l_t}\right) - 1\right)}\right)(V_{bi} - \Phi_s) \qquad \textbf{(94)}$$

Drain-induced barrier lowering is modeled the same way, the threshold voltage shift due to DIBL is calculated as:

$$\Delta V_{th}(DIBL) = -\frac{(ETA0 + ETAB \cdot V_{bs})}{2\left(\cosh\left(DSUB \cdot \frac{L_{eff}}{l_{t0}}\right) - 1\right)} V_{ds} \qquad \textbf{(95)}$$

DVT1 is basically equal to $1/(\eta)^{1/2}$, ETAB and DVT2 represent the influence of substrate bias effects on SCE and DIBL.

Narrow-Width Effect

The existence of fringing fields leads to a depletion region in the channel that is always larger as is calculated using one-dimensional analysis. This effect gains more influence with decreasing channel widths since the depletion region underneath the fringing field becomes comparable to the depletion field formed in vertical direction. The result is an increase of V_{th}.

The formulation for the narrow-width effect is:

$$\Delta V_{th} = (K3 + K3B \cdot V_{bs}) \cdot \frac{TOXE}{W'_{eff} + W0} \cdot \Phi_s \tag{96}$$

$$- \left(\frac{DVT0W}{2\left(\cosh\left(DVT1W \cdot \frac{L_{eff} \cdot W'_{eff}}{l_{tw}} \right) - 1 \right)} \cdot (V_{bi} - \Phi_s) \right)$$

$$W'_{eff} = \frac{W_{drawn}}{NF}$$

Subthreshold Swing

In the subthreshold region, the drain current flow is modeled by:

$$I_d = \mu \cdot \frac{W}{L} \cdot \sqrt{\frac{q \cdot \varepsilon_{Si} \cdot NDEP}{2 \cdot \Phi_S}} \cdot v_t^2 \cdot \left[1 - \exp\left(-\frac{V_{ds}}{v_t} \right) \right] \cdot \tag{97}$$

$$\exp\left(\frac{V_{gs} - V_{th} - VOFF - (VOFFL / L_{eff})}{n \cdot v_t} \right)$$

where $v_t = \frac{k_B \cdot T}{q}$ is the thermal voltage.

The expression $VOFF + \frac{VOFFL}{L_{eff}}$ represents the offset voltage and

gives the channel current at $V_{gs} = 0$.

The subthreshold swing parameter n is determined by channel length and interface state density and is calculated using

$$n = 1 + NFACTOR \cdot \frac{C_{dep}}{C_{oxe}} + \left[\frac{\frac{1}{2}(CDSC + CDSCD \cdot V_{ds} + CDSCB \cdot V_{bseff})}{\cosh\left(DVT1 \cdot \frac{L_{eff}}{lt}\right) - 1} + CIT\right] / C_{oxe} \quad \textbf{(98)}$$

The parameter NFACTOR is used to compensate for errors in calculating the depletion width capacitance.

Table 28 Subthreshold swing parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
W	W	channel width	0.25E-6
L	L	channel length	5E-6
μ		carrier mobility	$\begin{bmatrix} electrons & 670 \\ holes & 270 \end{bmatrix}$ cm²/Vsec
VOFF	VOFF	Offset voltage in subthreshold region for large W and L	-0.08 mV
VOFFL	VOFFL	Channel length dependence of VOFF	0 V
NFACTOR	NFACTOR	Subthreshold swing factor	1.0

Table 28 Subthreshold swing parameters (continued)

Equation Variable	BSIM4 Parameter	Description	Default Value
CIT	CIT	Interface trap capacitance	0 F/m^2
CDSC	CDSC	Drain-Source to channel coupling capacitance	2.4E-4 F/m^2
CDSCB	CDSCB	Body-bias coefficient of CDSC	0 F/Vm^2
CDSCD	CDSCD	Drain-bias coefficient of CDSC	0 F/Vm^2
C_{dep}		depletion capacitance	

Gate Direct Tunneling Current Model

Gate oxide thickness is decreasing, therefore tunneling currents from the gate contact are playing an important role in the modeling of sub micrometer MOSFET's. In BSIM4, the gate current consists of one part tunneling from gate to bulk (I_{gb}) and one part tunneling from gate to channel (I_{gc}). The latter one again is partitioned to flow to the source contact (I_{gcs}) and to the drain contact (I_{gcd}), as well as from the gate to the source and drain diffusion regions (I_{gs}, I_{gd}), as is shown in the following figure.

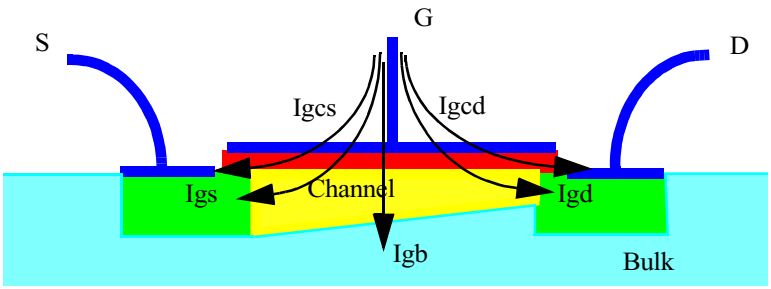


Figure 159 Cross section of a MOSFET with gate tunneling current components

Two model selectors are used to turn on or off tunneling current components, IGBMOD and IGCMOD. Setting IGBMOD = 1 turns on I_{gb}, IGCMOD = 1 turns on I_{gc}, I_{gs} and I_{gd}. Setting IGBMOB = IBGCMOD = 0 turns off modeling of gate tunneling currents.

The BSIM4.3.0 Version of the model allows the modeling of Gate Current Tunneling through Multiple-Layer Stacks by use of a tunneling attenuation coefficient.

Gate-to-Bulk Current

This current consists of two parts, tunneling of *electrons* from the *conduction* band and from the *valence* band. The first part is significant in the accumulation region, the second one during device inversion. The accumulation region tunneling current is dominated by electron tunneling from the conduction band and is given by:

$$I_{gbacc} = W_{eff} \cdot L_{eff} \cdot A \cdot \left(\frac{TOXREF}{TOXE}\right)^{NTOX} \cdot \frac{1}{TOXE^2} \cdot V_{gb} \quad \textbf{(99)}$$

$$\cdot V_{aux1} \cdot \exp[-B \cdot TOXE(AIGBACC - BIGBACC \cdot V_{oxacc}) \cdot$$

$$(1 + CIGBACC \cdot V_{oxacc})]$$

Inside this equation, the auxiliary voltage is:

$$V_{aux1} = NIGBACC \cdot (v_t) \quad \textbf{(100)}$$

$$\cdot \log\left(1 + \exp\left(-\frac{V_{gb} - V_{fbzb}}{NIGBACC \cdot v_t}\right)\right)$$

The constants in this equation are:

A = 4.97232E-7 A/V^2

B = 7.45669E11 $(g/F\text{-}s^2)^{0.5}$

The tunneling current dominating the inversion region is caused by electron tunneling from the valence band. It is calculated by:

$$I_{gbinv} = W_{eff} \cdot L_{eff} \cdot C \cdot \left(\frac{TOXREF}{TOXE}\right)^{NTOX} \cdot \frac{1}{TOXE^2} \cdot V_{gb} \quad \textbf{(101)}$$

$$\cdot V_{aux2} \cdot \exp[-D \cdot TOXE(AIGBINV - BIGBINV \cdot V_{oxdepinv})$$

$$\cdot (1 + CIGB(ACCINV) \cdot V_{oxdepinv})]$$

Inside this equation, the auxiliary voltage is:

$$V_{aux2} = NIGBINV \cdot v_t \cdot \log\left(1 + \exp\left(\frac{V_{oxdepinv} - EIGBINV}{EIGBINV \cdot v_t}\right)\right)$$

The constants in the above equation are:

C= 3.75956E-7 A/V²

D= 9.82222E11 $(g/F\text{-}s^2)^{0.5}$

The voltage across the gate oxide V_{ox} consists of the oxide voltage in accumulation and the one in inversion, as used in Equation 100 and Equation 101.

$$V_{ox} = V_{oxacc} + V_{oxdepinv} \quad \textbf{(102)}$$

The parts of V_{ox} are calculated by:

$$V_{oxacc} = -\frac{1}{2} \cdot \left[\left(V_{fbzb} - V_{gb} - \frac{2}{100}\right) + \sqrt{\left(V_{fbzb} - V_{gb} - \frac{2}{100}\right)^2 + \frac{8}{100} \cdot V_{fbzb}}\right]$$

$$V_{oxdepinv} = K_{1ox} \cdot \sqrt{\Phi_s} + V_{gsteff}$$

The flatband voltage, calculated from zero-bias V_{th} is:

$$V_{fbzb} = V_{th}\Big|_{V_{bs}andV_{ds}=0} - \Phi_s - K1 \cdot \sqrt{\Phi_s}$$

Equation 102 is continuously valid from accumulation through depletion to inversion.

Table 29 Gate Tunneling Parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
TOXREF	TOXREF	Nominal gate oxide thickness for gate direct tunneling model	3E-9 m
NTOX	NTOX	Exponent for the gate oxide ratio	1.0
AIGBACC	AIGBACC	Parameter for I_{gb} in accumulation	0.43
BIGBACC	BIGBACC	Parameter for I_{gb} in accumulation	0.054
CIGBACC	CIGBACC	Parameter for I_{gb} in accumulation	0.075
NIGBACC	NIGBACC	Parameter for I_{gb} in accumulation	1.0
AIGBINV	AIGBINV	Parameter for I_{gb} in inversion	0.35
BIGBINV	BIGBINV	Parameter for I_{gb} in inversion	0.03
CIGBINV	CIGBINV	Parameter for I_{gb} in inversion	0.006
NIGBINV	NIGBINV	Parameter for I_{gb} in inversion	1.1

Gate-to-Channel Current I_{gc}

The gate-to-channel current is determined by electrons tunneling from the conduction band in NMOS transistors respective holes tunneling from the valence band in PMOS transistors.

$$I_{gc} = W_{eff} \cdot L_{eff} \cdot E \cdot \left(\frac{TOXREF}{TOXE}\right)^{NTOX} \cdot \frac{1}{TOXE^2} \cdot V_{gse} \quad \textbf{(103)}$$
$$\cdot NIGC \cdot v_t \cdot \log\left(1 + \exp\left(\frac{V_{gse} - VTH0}{NIGC \cdot v_t}\right)\right)$$
$$\cdot \exp[-F \cdot TOXE \cdot (AIGC - BIGC \cdot V_{oxdepinv})$$
$$\cdot (1 + CIGC \cdot V_{oxdepinv})]$$

The physical constants E and F are listed in Table 30.

Gate-to-Source and Gate-to-Drain tunneling currents

These currents tunnel from the gate contact to the source or drain diffusion regions. They are caused by electron tunneling from the conduction band in NMOS transistors and by hole tunneling from the valence band in PMOS transistors.

$$I_{gs} = W_{eff} \cdot DLCIG \cdot E \cdot T_{oxRatioEdge} \cdot V_{gs} \quad \textbf{(104)}$$
$$\cdot V_{gs}' \cdot \exp[-F \cdot TOXE \cdot POXEDGE$$
$$\cdot (AIGSD - (BIGSD \cdot V_{gs}')) \cdot (1 + CIGSD \cdot V_{gs}')]$$

$$V_{gs}' = \sqrt{(V_{gs} - V_{fbsd})^2 + (1e-4)}$$

For the computing of I_{gd}, the values of V_{gs} in Equation 104 has to be replaced by V_{gd}.

The flat-band voltage between and the source or drain diffusion areas is dependent from NGATE:

If NGATE > 0.0:

$$V_{fbsd} = \frac{k_B T}{q} \cdot \log\left(\frac{NGATE}{NSD}\right)$$

Else:

$$V_{fbsd} = 0$$

To take drain bias effects into account, the tunneling current from the gate contact splits into two components and it is $I_{gc} = I_{gcs} + I_{gcd}$. The components are calculated as:

$$I_{gcs} = I_{gc} \cdot \frac{PIGCD \cdot V_{ds} + \exp(-PIGCD \cdot V_{ds}) - 1 + (1e-4)}{PIGCD^2 \cdot V_{ds}^2 + (2e-4)}$$

$$I_{gcd} = I_{gc} \cdot \frac{1-(PIGCD \cdot V_{ds} + 1)\exp(-PIGCD \cdot V_{ds}) + (1e-4)}{PIGCD^2 \cdot V_{ds}^2 + (2e-4)}$$

If the model parameter PIGCD is not specified, it is calculated by:

$$PIGCD = \frac{B \cdot TOXE}{V_{gsteff}^2} \cdot \left(1 - \frac{V_{dseff}}{2 \cdot V_{gsteff}}\right) \quad \textbf{(105)}$$

The constants used in Equation 104 and Equation 105 have different values for NMOS and PMOS transistors:

Table 30 Values of constants for gate-channel and gate S/D tunneling

	NMOS	PMOS
E	4.97232 A/V²	3.42537 A/V²
F	7.45669E11 $\sqrt{g/(F-s^2)}$	1.16645E12 $\sqrt{g/(F-s^2)}$

Table 31 Gate Tunneling Parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
DLCIG	DLCIG	Source/Drain overlap length for I_{gs} and I_{gd}	LINT
POXEDGE	POXEDGE	Factor for gate oxide thickness in source/drain overlap regions	1.0

Table 31 Gate Tunneling Parameters (continued)

Equation Variable	BSIM4 Parameter	Description	Default Value
AIGSD	AIGSD	Parameter for I_{gs} and I_{gd}	NMOS: 0.43 PMOS: 0.31 $\left[\frac{\sqrt{(Fs^2)/g}}{m}\right]$
BIGSD	BIGSD	Parameter for I_{gs} and I_{gd}	NMOS: 0.054 PMOS: 0.024 $\left[\frac{\sqrt{(Fs^2)/g}}{m \cdot V}\right]$
CIGSD	CIGSD	Parameter for I_{gs} and I_{gd}	NMOS: 0.075 PMOS: 0.03 [V]
NIGC	NIGC	Parameter for I_{gcs}, I_{gcd}, I_{gs} and I_{gd}	1.0
PIGCD	PIGCD	Vgs dependence of I_{gcs} and I_{gcd}	1.0

Drain Current Model

Bulk Charge

If a drain-source voltage other than zero volts is applied, the depletion width along the channel will not be uniform. Therefore, the threshold voltage VTH will vary along the channel. This phenomenon is known as the *Bulk Charge Effect*. Inside BSIM4, the bulk charge effect is formulated as follows:

$$A_{bulk} = \frac{1}{1 + KETA \cdot V_{bseff}} \cdot \left\{ 1 + \left(\frac{\sqrt{1 + LPEB / L_{eff}} \cdot K_{1OX}}{2 \cdot \sqrt{\Phi_s - V_{bseff}}} + K_{2OX} - \left(K3B \cdot \frac{TOXE}{W_{eff}' + W0} \cdot \Phi_s \right) \right) \cdot \left[\frac{A0 \cdot L_{eff}}{\left(1 - AGS \cdot V_{gsteff} \cdot \left(\frac{L_{eff}}{L_{eff} + 2\sqrt{XJ \cdot X_{dep}}} \right)^2 \right) + \frac{B0}{W_{eff}' + B1}} \right] \right\} \quad \textbf{(106)}$$

NOTE A_{bulk} is about 1 for small channel lengths and increases with increasing channel length.

Table 32 Drain Current Parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
A_{bulk}		Bulk charge effect	
A0	A0	Bulk charge effect coefficient	1.0
XJ	XJ	Source/Drain junction depth	150E-9 m
X_{dep}		depletion depth	
AGS	AGS	Coefficient of V_{gs} dependence of bulk charge effect	0.0 1/V
B0	B0	Bulk charge effect coeff. for channel width	0.0 m
B1	B1	Bulk charge effect width offset	0.0 m
KETA	KETA	Body-bias coefficient of the bulk charge effect	-0.047 1/V

Unified Mobility Model

Mobility of carriers depends on many process parameters and bias conditions. Modeling mobility accurately is critical to precise modeling of MOS transistors. BSIM4 provides three different mobility models, selectable through the MOBMOD flag. The MOBMOD = 0 and 1 models are the same as being used in BSIM3v3. There is a new and accurate universal mobility model, selectable through MOBMOD = 2, which is also suitable for predictive modeling [1].

MOBMOD = 0:

(107)

$$\mu_{eff} = \frac{U0}{1 + (UA + UC \cdot V_{bseff}) \cdot \left(\frac{V_{gsteff} + 2V_{th}}{TOXE}\right) + UB \cdot \left(\frac{V_{gsteff} + 2V_{th}}{TOXE}\right)^2}$$

MOBMOD = 1:

(108)

$$\mu_{eff} = \frac{U0}{1 + \left[UA \cdot \left(\frac{V_{gsteff} + 2V_{th}}{TOXE}\right) + UB \cdot \left(\frac{V_{gsteff} + 2V_{th}}{TOXE}\right)^2\right] \cdot (1 + UC \cdot V_{bseff})}$$

MOBMOD = 2

(109)

$$\mu_{eff} = \frac{U0}{1 + (UA + UC \cdot V_{bseff})\left(\frac{V_{gsteff} + C_0 \cdot (VTH0 - VFB - \Phi_s)}{TOXE}\right)^{EU}}$$

The constant C_0 has different values for different MOS processes. For NMOS processes C_0 = 2 is used and for PMOS processes C_0 = 2.5 is used.

Table 33 Mobility Model Parameters

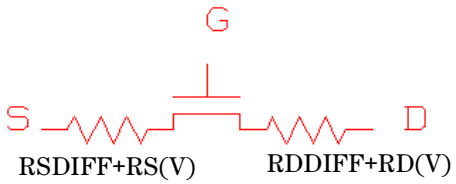
Equation Variable	BSIM4 Parameter	Description	Default Value
U0	U0	Low-field mobility	NMOS: 670 PMOS: 250 [cm²/(Vs)]
UA	UA	First-order mobility degradation coefficient due to vertical field	MOBMOD=0, 1: 1E-9 MOBMOD=2: 1e-15 [m/V]
UB	UB	Second-order mobility degradation coefficient	1E-19 (m/V)²
UC	UC	Coefficient of the body-bias effect of mobility degradation	MOBMOD=1: -0.0465 1/V; MOBMOD=0, 2: 0.0465E-9 m/V^2
EU	EU	Exponent for mobility degradation of MOBMOD = 2	NMOS: 1.67 PMOS: 1.0

Drain / Source Resistance Model

The resistances of the drain/source regions are modeled using two components: The sheet resistance, which is bias-independent, and a bias-dependent LDD resistance.

In contrast to the BSIM3 models, the drain and source LDD resistances are not necessarily the same, they could be asymmetric. This is a prerequisite for accurate RF simulations.

A further enhancement of the BSIM4 model over BSIM3 is the external or internal RDS option, invoked by the model selector RDSMOD = 0 (internal RDS) or RDSMOD = 1 (external RDS). The external RDS option looks at a resistance connected between the internal and external source and drain nodes. See the following figure.



RDSMOD = 0 (internal $R_s(V)$)

$$R_{ds}(V) = \frac{1}{(1e6 \cdot W_{effcj})^{WR}} \left\{ \frac{RDSWMIN + RDSW}{\left[PRWB \cdot (\sqrt{\Phi_s - V_{bseff}} - \sqrt{\Phi_s}) + \frac{1}{1 + PRWG \cdot V_{gsteff}} \right]} \right\} \quad \textbf{(110)}$$

RDSMOD = 1 (external $R_d(V)$ and $R_s(V)$)

$$R_d(V) = \frac{1}{(1e6 \cdot W_{effcj})^{WR} \cdot NF} \left\{ \frac{RDSWMIN + RDSW}{\left[-PRWB \cdot V_{bd} + \frac{1}{1 + PRWG \cdot (V_{gd} - V_{fbsd})} \right]} \right\} \quad (111)$$

$$R_s(V) = \frac{1}{(1e6 \cdot W_{effcj})^{WR} \cdot NF} \left\{ \frac{RDSWMIN + RDSW}{\left[-PRWB \cdot V_{bs} + \frac{1}{1 + PRWG \cdot (V_{gs} - V_{fbsd})} \right]} \right\} \quad (112)$$

The flatband voltage V_{fbsd} is calculated as follows:

If NGATE > 0:

$$V_{fbsd} = \frac{k_B \cdot T}{q} \cdot \log\left(\frac{NGATE}{NSD}\right)$$

Else:

$$V_{fbsd} = 0$$

Table 34 Drain Source Resistance Parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
NGATE	NGATE	Poly Si-gate doping concentration	0.0 cm^{-3}
PRWB	PRWB	Body bias coefficient of LDD resistance	0.0 $V^{-0.5}$

Table 34 Drain Source Resistance Parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
PRWG	PRWG	Gate bias dependence of LDD resistance	1.0 1/V
RDSW	RDSW	Zero bias LDD resistance per unit width for RDSMOD = 0	200 Ω (µm)^{WR}
RDSWMIN	RDSWMIN	LDD resistance per unit width at high V_{gs} and zero V_{bs} for RDSMOD = 0	0.0 Ω (µm)^{WR}
WR	WR	Channel width dependence parameter of LDD resistance	1.0

Saturation Region Output Conductance Model

The following figure shows a typical MOSFET I_{ds} vs. V_{ds} diagram. The calculated output resistance is inserted into the diagram as well. This output resistance curve can be divided into four distinct regions, each region is affected by different physical effects. The first region at low Vds is characterized by a very small output resistance. It is called the linear region, where carrier velocity is not yet saturated.

Increasing Vds leads to a region that is dominated by carrier velocity saturation—this is the so-called saturation region. In this region, three different physical mechanisms are controlling device behavior. Those mechanisms are Channel Length Modulation (CLM), Drain-Induced Barrier Lowering (DIBL), and Substrate-Current Induced Body Effect (SCBE). Each of those mechanisms dominate the output resistance in a specific region.

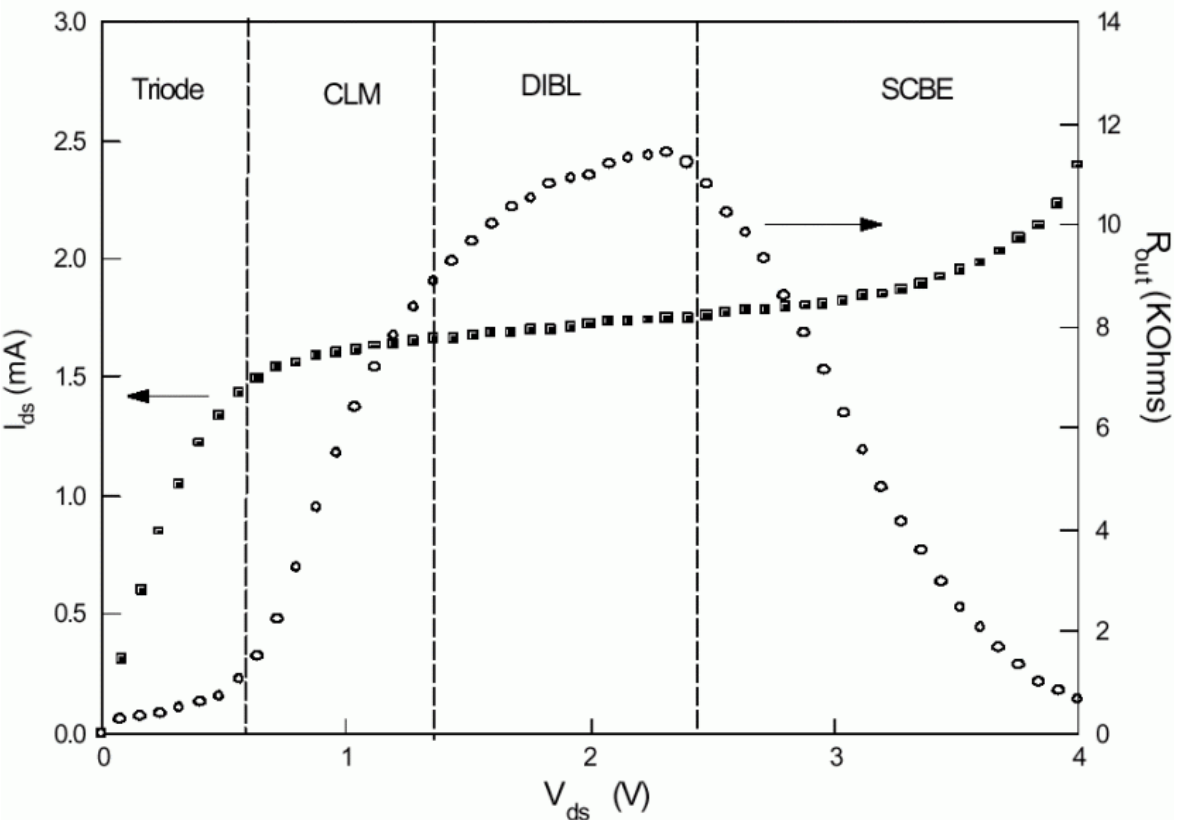


Figure 160 Output Resistance vs. Drain-Source Voltage [1]

The continuous channel current equation for the linear and saturation region as implemented in BSIM4 is:

$$I_{ds} = \left(\frac{I_{ds0} \cdot NF}{1 + \frac{R_{ds} \cdot I_{ds0}}{V_{dseff}}} \right) \left(\left[1 + \frac{1}{C_{clm}} \cdot \ln \left(\frac{V_{Asat} + V_{ACLM}}{V_{Asat}} \right) \right] \right. \quad \textbf{(113)}$$

$$\left. \cdot \left(1 + \frac{V_{ds} - V_{dseff}}{V_{ADIBL}} \right) \left(1 + \frac{V_{ds} - V_{dseff}}{V_{ADITS}} \right) \right) \cdot \left(1 + \frac{V_{ds} - V_{dseff}}{V_{ASCBE}} \right)$$

The Early voltage V_{Asat} at $V_{ds} = V_{dsat}$ is used to get continuous expressions for drain current and output resistance between linear and saturation region.

$$V_{Asat} = \left[1 - \frac{A_{bulk} \cdot V_{dsat}}{2 \cdot \left(V_{gsteff} + 2 \frac{k_B \cdot T}{q} \right)} \right] \quad \textbf{(114)}$$

$$\cdot \frac{E_{sat} \cdot L_{eff} + V_{dsat} + 2 \cdot R_{ds} \cdot vsat \cdot C_{oxe} \cdot W_{eff} \cdot V_{gsteff}}{R_{ds} \cdot vsat \cdot C_{oxe} \cdot W_{eff} \cdot A_{bulk} - 1 + \frac{2}{\lambda}}$$

In this equation, the channel current dependencies are modeled using specific Early voltages V_A, as will be described in the following sections.

Channel Length Modulation - CLM

Through integration based on a quasi two-dimensional analysis, we obtain

$$V_{ACLM} = \frac{1}{PCLM} \cdot \frac{(V_{ds} - V_{dsat})}{1 + FPROUT \cdot \frac{\sqrt{L_{eff}}}{V_{gsteff} + 2 \cdot \frac{k_B \cdot T}{q}}} \quad (115)$$

$$\cdot \left(1 + PVAG \cdot \frac{V_{gsteff}}{E_{sat} \cdot L_{eff}}\right)$$

$$\cdot \left(1 + \frac{R_{ds} \cdot I_{dso}}{V_{dseff}}\right) \cdot \left(L_{eff} + \frac{V_{dsat}}{E_{sat}}\right) \cdot \sqrt{\frac{\varepsilon_{si} \cdot TOXE \cdot XJ}{EPSROX}}$$

Drain-Induced Barrier Lowering - DIBL

The gate voltage modulates the DIBL effect. To correctly model DIBL, the parameter PDIBLC2 is introduced. This parameter becomes significant only for long channel devices.

$$V_{ADIBL} = \frac{V_{gsteff} + 2\nu_t}{\theta_{rout} \cdot (1 + PDIBLCB \cdot V_{bsteff})} \quad (116)$$

$$\cdot \left(1 - \frac{A_{bulk} \cdot V_{dsat}}{A_{bulk} \cdot V_{dsat} + V_{gsteff} + 2\frac{k_B \cdot T}{q}}\right)$$

$$\cdot \left(1 + PVAG \cdot \frac{V_{gsteff}}{E_{sat} \cdot L_{eff}}\right)$$

The parameter θ_{rout} is channel length dependent in the same manner as the DIBL effect in V_{TH}, but different parameters are used here.

$$\theta_{rout} = \frac{PDIBLC1}{2 \cdot \cosh\left(\frac{DROUT \cdot L_{eff}}{lt0}\right) - 2} + PDIBLC2$$

Substrate-Current Induced Body Effect - SCBE

Due to increasing V_{ds}, some electrons flowing from the source of an NMOS device will gain high energies and are able to cause impact ionization. Electron-hole pairs will be generated and the substrate current created by impact ionization will increase exponentially with the drain voltage. The early voltage due to SCBE is calculated by

$$V_{ASCBE} = \frac{L_{eff}}{PSCBE2 \cdot \exp\left(-\frac{PSCBE1}{V_{ds} - V_{dsat}}\right)} \quad \textbf{(117)}$$

If the device is produced using pocket implantation, a potential barrier at the drain end of the channel will be introduced. The potential barrier can be reduced by the drain voltage even in long channel devices. This effect is called Drain-Induced Threshold Shift (DITS) and the early voltage due to DITS is

$$V_{ADITS} = \frac{1}{PDITS} \cdot \frac{1}{1 + FPROUT \cdot \frac{\sqrt{L_{eff}}}{V_{gsteff} + 2 \cdot \frac{k_B \cdot T}{q}}} \cdot \quad \textbf{(118)}$$

$$\{1 + [(1 + PDITSL \cdot L_{eff}) \cdot \exp(PDITSD \cdot V_{ds})]\}$$

Table 35 Saturation Region Output Conductance Parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
DROUT	DROUT	Channel-length dependence coefficient of the DIBL effect on output resistance	0.56
PSCBE1	PSCBE1	First substrate current induced body-effect parameter	4.24E8 V/m
PSCBE2	PSCBE2	Second substrate current induced body-effect coefficient	1.0E-5 m/V
PVAG	PVAG	Gate-bias dependence of Early voltage	0.0
FPROUT	FPROUT	Effect of pocket implant on R_{out} degradation	0.0 $V/m^{0.5}$
PDITS	PDITS	Impact of drain-induced V_{th} shift on R_{out}	0.0 V^{-1}
PDITSL	PDITSL	Channel-length dependence of drain-induced V_{th} shift on R_{out}	0.0
PDITSD	PDITSD	V_{ds} dependence of drain-induced V_{th} shift on R_{out}	0.0V^{-1}
PCLM	PCLM	Channel length modulation parameter	1.3
PDIBLC1	PDIBLC1	First output resistance DIBL effect parameter	0.39
PDIBLC2	PDIBLC2	Second output resistance DIBL effect parameter	8.6m
PDIBLCB	PDIBLCB	Body bias coefficient of output resistance DIBL effect	0.0 1/V

Body Current Model

The substrate current of a MOSFET consists of diode junction currents, gate-to-body tunneling current, impact-ionization (I_{ii}) and gate-induced drain leakage currents (I_{GIDL}).

Impact Ionization Model

BSIM4 uses the same impact ionization model as was introduced in BSIM3v3.2. The impact ionization current is calculated by

$$I_{ii} = \frac{ALPHA0 + ALPHA1 \cdot L_{eff}}{L_{eff}} \cdot (V_{ds} - V_{dseff}) \qquad \textbf{(119)}$$

$$\cdot \exp\left(\frac{BETA0}{V_{ds} - V_{dseff}}\right) \cdot \frac{I_{ds0} \cdot NF}{1 + \frac{R_{ds} \cdot I_{ds0}}{V_{dseff}}} \cdot \left[1 + \frac{1}{C_{clm}} \cdot \ln\frac{V_A}{V_{Asat}}\right]$$

$$\cdot \left(1 + \frac{V_{ds} - V_{dseff}}{V_{ADIBL}}\right) \cdot \left(1 + \frac{V_{ds} - V_{dseff}}{V_{ADITS}}\right)$$

Gate-Induced Drain Leakage

The GIDL effect is modeled by

$$I_{GIDL} = AGIDL \cdot W_{effCJ} \cdot NF \cdot \frac{V_{ds} - V_{gse} - EGIDL}{3 \cdot TOXE} \cdot \exp\left(-\frac{3 \cdot TOXE \cdot BGIDL}{V_{ds} - V_{gse} - EGIDL}\right) \cdot \frac{V_{db}^3}{CGIDL + V_{db}^3} \quad (120)$$

Table 36 Body Current Model Parameters

Equation Variable	Parameter Name	Description	Default Value
ALPHA0	ALPHA0	First impact ionization parameter	0.0 Am/V
ALPHA1	ALPHA1	Length dependent substrate current parameter	0.0 A/V
BETA0	BETA0	Second impact ionization parameter	30 V
AGIDL	AGIDL	Pre-exponential coefficient for GIDL	0.0 mho (1/Ohm)
BGIDL	BGIDL	Exponential coefficient for GIDL	2.3e9 V/m
CGIDL	CGIDL	Parameter for body-bias effect on GIDL	0.5 V^3
EGIDL	EGIDL	Fitting parameter for band bending for GIDL	0.8 V

Stress Effect Modeling

The scaling of CMOS feature sizes makes shallow trench isolation (STI) a popular technology. To enhance device performance, strain channel materials have been used. The mechanical stress introduced by using these processes causes MOSFET performance to become a function of the active device area and the location of the device in the isolated region. Influence of stress on mobility and saturation velocity has been known since the 0.13 um technology [1].

For the named reasons, BSIM4 considers the influence of stress on:

- mobility

- velocity saturation
- threshold voltage
- body effect
- DIBL effect.

Mobility related dependence of device performance is induced through band structure modification. Doping profile variation results in Vth dependence of the stress effect. Both effects follow the same 1/LOD trend but have different L and W scaling influence. By modifying some parameters in the BSIM model, a phenomenological model has been implemented. The model assumes mobility relative change to be proportional to stress distribution.

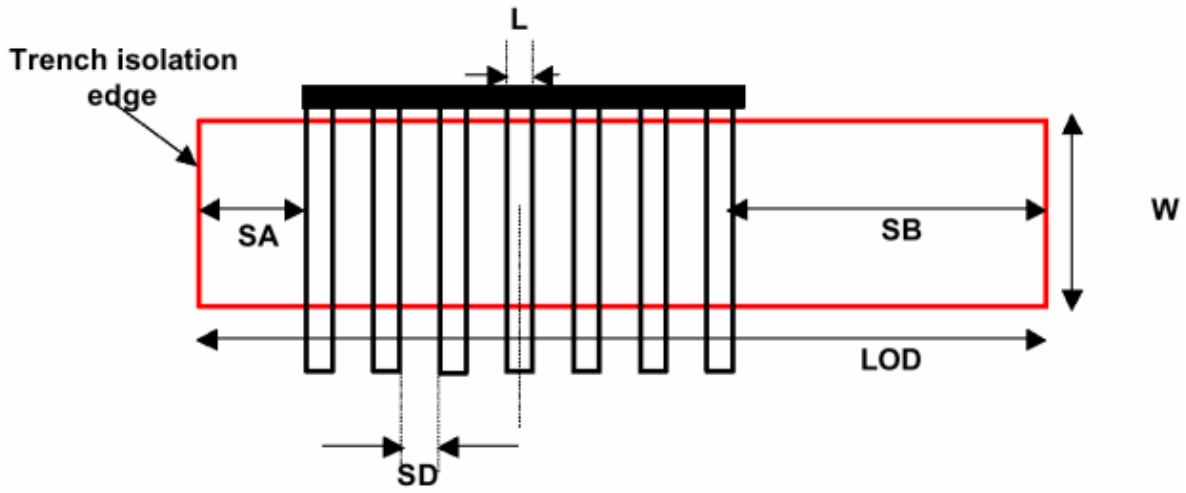


Figure 161 MOSFET device geometry using a shallow trench isolation scheme

The figure above shows a typical MOSFET layout surrounded by shallow trench isolation. SA, SB are the distances between trench isolation edge to Gate-PolySi from one and from the other side, respectively. SD is the distance between neighboring fingers of the device. The **L**ength of **O**xide **D**efinition (LOD) is expressed through the following equation:

$$LOD = SA + SB + N \cdot L + (N - 1) \cdot SD$$

2D simulation shows that stress distribution can be expressed by a simple function of SA and SB.

To cover doping profile changes in devices with different LOD; Vth0, K2, and ETA0 are modified.

The total LOD effect for multiple finger devices is the average of the LOD effect on every finger.

Since MOSFETs often use an irregular shape of their active area, additional instance parameters have to be introduced to fully describe the shape of the active area. This will result in many new parameters in the netlists and an increase in simulation time. To avoid this drawbacks, BSIM4.3.0 uses effective SA and SB values.

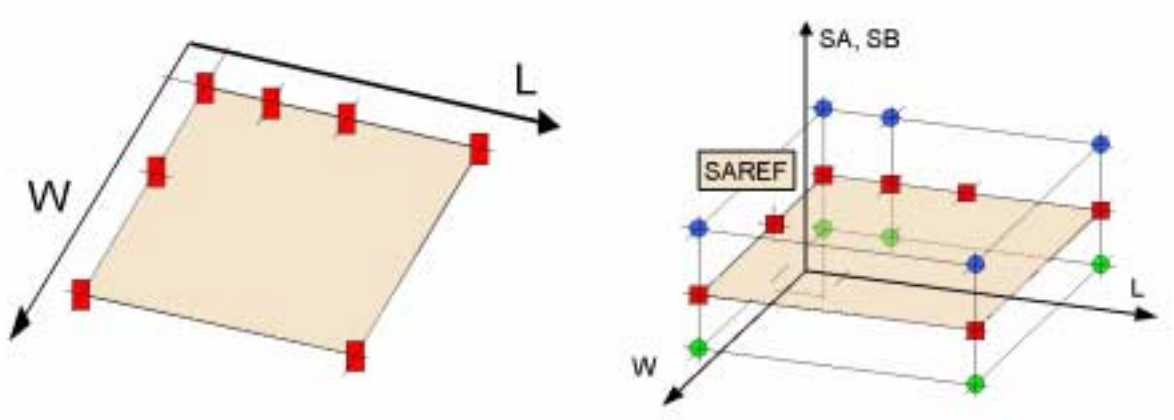


Figure 162 A “third” dimension is added to standard geometry parameters of MOSFET devices by introducing new parameters SA and SB to model stress effect influence on device performance

The left part of the figure above shows the geometry parameters used in MOSFET models so far. To the right, the SAREF-plane represents the standard L, W plane which is varied by SA and SB.

Until BSIM4.2.1, gate length (L) and gate width (W) have been the major device geometry parameters required.

For more details regarding the modeled stress effect influences on device performance, see Chapter 13 of the BSIM4.3.0 manual[1].

CV Modeling

Capacitance Model

To accurately model MOSFET behavior, a good capacitance model considering intrinsic and extrinsic (overlap/fringing) capacitances is important.

BSIM4 provides three options to select different capacitance models. These are the models from BSIM3v3.2, which are taken without changes. There is only one exception: Different parameters for source and drain sides are introduced, which are used to precisely model different doping concentrations and so on.

The model flag CAPMOD allows three values. CAPMOD = 0 uses piece-wise and simple equations, whereas with CAPMOD = 1 and 2 uses smooth and single equation models.

For CAPMOD = 0, VTH is taken from a long-channel device; for CAPMOD = 1 and 2, VTH is consistent with the BSIM4 DC model.

The overlap capacitance model uses a bias-independent part to model the effective overlap capacitance between gate and heavily doped source/drain regions and a gate-bias dependent part between the gate and the lightly doped source/drain regions.

Fringing capacitances between gate and source as well as gate and drain are modeled bias-independent.

Intrinsic Capacitance Modeling

All capacitances in Intrinsic Capacitance Model formulations are derived from terminal charges instead of terminal voltages to ensure charge conservation.

Long channel device models assume the mobility to be constant and no channel length modulation occurs. However, with shrinking device dimensions, velocity saturation and channel length modulation are to be considered to accurately model device behavior.

For capacitance modeling in BSIM4, a drain bias is defined, at which the channel charge becomes constant.

$$V_{dsat,CV} = \frac{NOFF \cdot n \cdot \frac{k_B \cdot T}{q}}{A_{bulk,CV} \cdot \left[1 + \left(\frac{CLC}{L_{active}}\right)^{CLE}\right]} \cdot \ln\left[1 + \exp\left(\frac{V_{gse} - V_{th} - VOFFCV}{NOFF \cdot n \cdot \frac{k_B \cdot T}{q}}\right)\right] \quad (121)$$

For capacitance modeling, A_{bulk} is defined different from DC:

$$A_{bulk,CV} = \left\{1 + \left[\frac{A0 \cdot L_{eff}}{L_{eff} + 2\sqrt{XJ \cdot X_{dep}}} + \frac{B0}{W_{eff}' + B1}\right]\right.$$

$$\left.\left(\frac{\sqrt{1 + LPEB/L_{eff}} \cdot K1OX}{2 \cdot \sqrt{\Phi_s - V_{bseff}}} + K2OX - K3B \cdot \frac{TOXE}{W_{eff}' + W0} \cdot \Phi_s\right)\right\}$$

$$\cdot \frac{1}{1 + KETA \cdot V_{bseff}}$$

Numerical simulation has shown that the charged layer under the gate of a MOSFET has a significant thickness in all regions of operation. Therefore, a Charge-Thickness Model has been introduced in BSIM4. This model uses a capacity in series with the oxide capacitance Cox and an effective oxide capacitance is used:

$$C_{oxeff} = \frac{C_{oxe} \cdot \frac{\varepsilon_{si}}{X_{DC}}}{C_{oxe} + \frac{\varepsilon_{si}}{X_{DC}}}$$

DC charge layer thickness in accumulation and depletion is calculated by:

$$X_{DC} = \frac{1}{3} \cdot L_{debye} \cdot \exp\left[ACDE \cdot \left(\frac{NDEP}{2 \cdot 10^{16}}\right)^{-1/4} \cdot \frac{V_{gse} - V_{bseff} - V_{FBeff}}{TOXE} \right] \quad (122)$$

whereas in inversion

$$X_{DC} = \frac{19 \cdot 10^{-10}}{1 + \left(\frac{V_{gsteff} + 4(VTH0 - VFB - \Phi_s)}{2 \cdot TOXP}\right)^{7/10}} \quad (123)$$

$$V_{FBeff} = V_{fbzb} - \frac{1}{2} \cdot \left[\left(V_{fbzb} - V_{gb} - \frac{2}{100}\right) + \sqrt{\left(V_{fbzb} - V_{gb} - \frac{2}{100}\right)^2 + \frac{8}{100} \cdot V_{fbzb}} \right]$$

$$V_{fbzb} = V_{th,0} - \Phi_S - K1 \cdot \sqrt{\Phi_S}$$

$$V_{th,0} = V_{th}\Big|_{V_{bs} and V_{ds} = 0}$$

By introducing the VFB term in Equation 123, the calculation is valid for N+ or P+ poly-silicon gates and future gate materials too.

Intrinsic Capacitance Model Equations

There are three intrinsic capacitance models to choose from using the model flag CAPMOD. Additionally, there are three charge partitioning schemes: 40/60, 50/50, and 0/100. Those schemes describe distribution of the intrinsic capacitance charges between drain and source side.

The exact formulations for the different operation regimes and charge partitioning schemes are in the UC Berkeley manual [1] on pages 7-13 to 7-19.

Fringing Capacitance Models

The fringing capacitance consists of a bias-independent outer fringing capacitance and an inner fringing capacitance, bias-dependent. The outer fringing capacitance is modeled in BSIM4 (if not given) through:

$$CF = \frac{2 \cdot EPSROX \cdot \varepsilon_0}{\pi} \cdot \log\left(1 + \frac{4 \cdot 10^{-7}}{TOXE}\right) \quad \textbf{(124)}$$

The inner capacitance is not modeled.

Overlap Capacitance Model

For accurate simulation results, especially the drain side overlap capacitance has to be modeled exactly because the influence of this capacitance is amplified by the gain of the transistor (Miller effect). Formerly used capacitance models assume a bias-independent overlap capacitance. However, experimental data show a gate-bias dependent overlap capacitance, which is invoked using CAPMOD = 1 or 2. Using CAPMOD = 0, a simple bias-independent model is invoked.

For CAPMOD = 0, the overlap charges are expressed by:

Gate-to-source overlap charge:

$$Q_{overlap,s} = W_{active} \cdot CGSO \cdot V_{gs}$$

Gate-to-drain overlap charge:

$$Q_{overlap,\,d} = W_{active} \cdot CGDO \cdot V_{gd}$$

Gate-to-bulk overlap charge:

$$Q_{overlap,\,b} = L_{active} \cdot CGBO \cdot V_{gb}$$

The parameters CGSO and CGDO are calculated (if not given) by:

Table 37 CGSO and CGDO Parameters

		CGSO	CGDO
If	DLC is given and > 0	$CGSO = DLC \cdot C_{oxe} - CGSL$ if CGSO < 0: CGSO = 0	$CGDO = DLC \cdot C_{oxe} - CGDL$ if CGDO < 0: CGDO = 0
Else		$CGSO = \frac{6}{10} \cdot XJ \cdot C_{oxe}$	$CGDO = \frac{6}{10} \cdot XJ \cdot C_{oxe}$

If CGBO is not given, it is calculated by:

$$CGBO = 2 \cdot DWC \cdot C_{oxe}$$

For CAPMOD = 1 or 2, the bias-dependent overlap charge is modeled at the source side by:

(125)

$$\frac{Q_{overlap,s}}{W_{active}} = (CGSO \cdot V_{gs}) + CGSL\left(V_{gs} - \frac{1}{2}\cdot\left(V_{gs} + \frac{2}{100} - \sqrt{\left(V_{gs} + \frac{2}{100}\right)^2 + \frac{8}{100}}\right) - \left(\frac{CKAPPAS}{2}\cdot\left(-1 + \sqrt{1 - \frac{2\cdot\left(V_{gs} + \frac{2}{100} - \sqrt{\left(V_{gs} + \frac{2}{100}\right)^2 + \frac{8}{100}}\right)}{CKAPPAS}}\right)\right)\right)$$

at the drain side by:

(126)

$$\frac{Q_{overlap,d}}{W_{active}} = (CGDO \cdot V_{gd}) + CGDL\left(V_{gd} - \frac{1}{2}\cdot\left(V_{gd} + \frac{2}{100} - \sqrt{\left(V_{gd} + \frac{2}{100}\right)^2 + \frac{8}{100}}\right) - \left(\frac{CKAPPAD}{2}\cdot\left(-1 + \sqrt{1 - \frac{2\cdot\left(V_{gd} + \frac{2}{100} - \sqrt{\left(V_{gd} + \frac{2}{100}\right)^2 + \frac{8}{100}}\right)}{CKAPPAD}}\right)\right)\right)$$

and the gate overlap charge by:

$$Q_{overlap,g} = -(Q_{overlap,d} + Q_{overlap,s}) - (CGBO \cdot L_{active}) \cdot V_{gb} \quad \textbf{(127)}$$

Table 38 Intrinsic Capacitance Model Parameters

Equation Variable	BSIM4 Parameter	Description	Default Value
NOFF	NOFF	CV parameter in $V_{gsteff,CV}$ for weak to strong inversion	1.0
VOFFCV	VOFFCV	CV parameter in $V_{gsteff,CV}$ for weak to strong inversion	0.0 V
ACDE	ACDE	Exponential coefficient for charge thickness in accumulation and depletion regions in CAPMOD=2	1.0 m/V
CKAPPAS	CKAPPAS	Coefficient of bias-dependent overlap capacitance on source side	0.6 V
CKAPPAD	CKAPPAD	Coefficient of bias-dependent overlap capacitance on drain side	CKAPPAS
CLC	CLC	Constant term for the short channel model	0.1E-7 m
CLE	CLE	Exponential term for the short channel model	0.6

Structure of the BSIM4 RF Simulation Model

The BSIM4 model consists of some major features that make it ideal for use in real high frequency simulations. It contains:

- scalable gate resistance
- dedicated thermal noise model formulation
- different device layouts (multifinger devices) are taken into account
- substrate resistance network with correct connection to the main transistor through parasitic diodes
- New in BSIM4.3.0: Horse-Shoe substrate contacts

While the first three effects are fully scalable, which means the influence of the device dimensions like gate length or width is already included in the model formulation, the substrate resistance effect included into BSIM4 has not included any of this information up to version 4.3.0. To specify the substrate resistance effect, only fixed values for up to 5 different resistors could have been set. However, in reality, the substrate resistance effect depends on sheet resistance of the body, number of gate fingers and width of the devices.

Due to this shortcoming, it is not possible to generate one fully scalable model card for a family of typical RF multifinger transistors. This is the reason why the BSIM4 Modeling Package includes two different approaches for generating BSIM4 RF models:

- Single transistor models describe the very classic approach, where one simulation model is generated for each available test device. A design library based on such models does not enable the circuit designer to modify major device dimensions. Only available devices can be used to design circuits. A major benefit of this approach is that the accuracy of such a model may be very high because only one certain device behavior has to be fitted by the model parameters.

- Scalable transistor models cover a certain range of major device dimensions. In the case of a RF MOS transistor, these are the gate length and gate width of a single transistor finger and the number of gate fingers. These models have a structure, such that a design engineer can change the parameters to get an optimum transistor behavior for a certain application.

In BSIM4.3.0, there is a newly introduced enhancement to modeling the substrate resistance. To take care of different geometric layouts, a so called “horseshoe” contact geometry was added (see “Substrate contact resistance scaling” on page 389).

The following two sections describe the structure of the single and the scalable model approaches. At first, the general structure of the BSIM4 RF model is shown.

Single Transistor Model

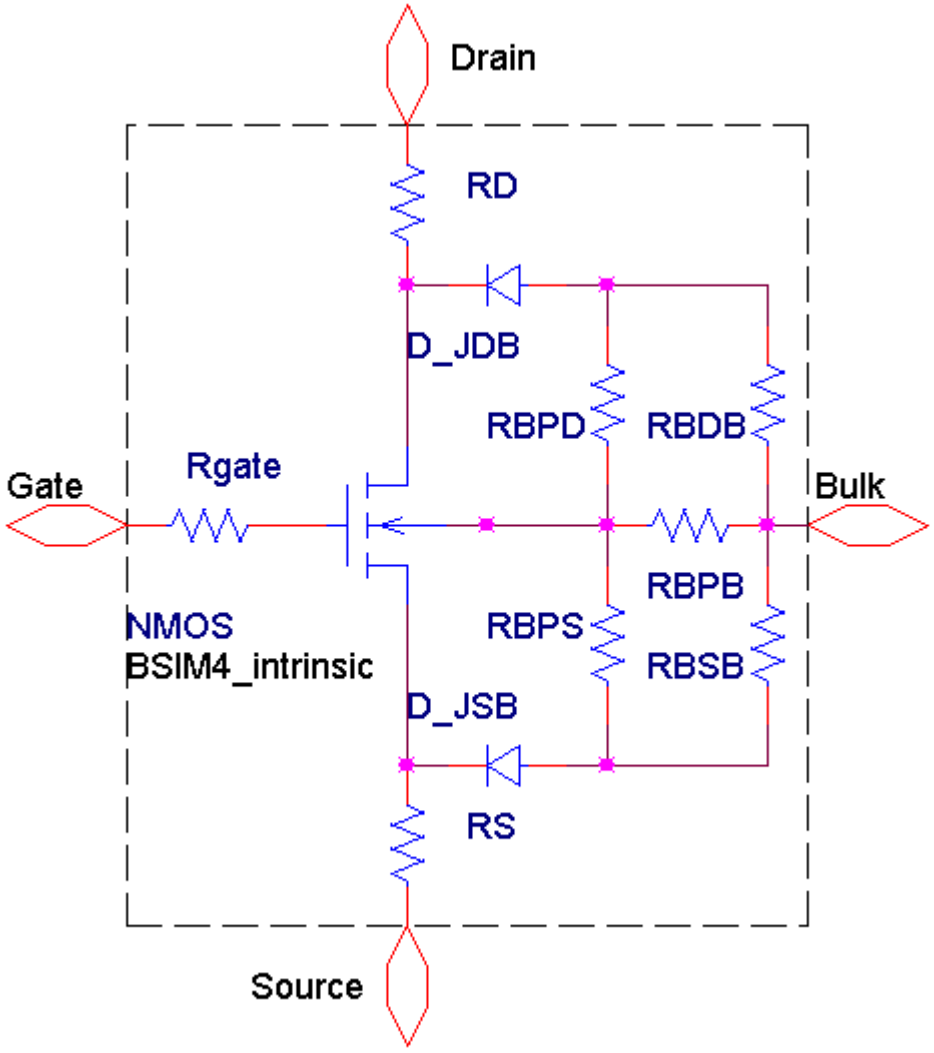


Figure 163 Schematic for the Single Transistor Model

For the single transistor model, all substrate resistance parameters RBPB, RBPD, RBPS, RBDB, and RBSB are set to fixed values for one certain device. This is the default approach, which is supported by the BSIM4 model. The description of the model in a simulator is very easy because only the call of a model card is necessary. The following netlist in spice3e2 syntax shows an example for this model description:

```
 .OPTIONS GMIN=1.0E-14 * *
-------------------------------------------------------------------------
* Model card for BSIM4.3.0 n-type devices | * | * Simulator: SPICE3e2 |
* Model: BSIM4 Modeling Package | * Date: 16.07.2004 | * Origin:
ICCAP_ROOT/..../bsim4/circuits/spice3/cir/nmos.cir | *
-------------------------------------------------------------------------
M1 1=D 2=G 3=S 4=B MOSMOD L=0.25u W=5u NF=1 AD=5p AS=5p PD=12u PS=12u
SA=0 SB=0 SD=0 NRD=0 NRS=0 .MODEL MOSMOD NMOS
LEVEL = 14 VERSION = 4.3.0 BINUNIT = 2 PARAMCHK = 1 + MOBMOD = 1
RDSMOD = 0 IGCMOD = 0 IGBMOD = 0 CAPMOD = 2 + RGATEMOD = 0 RBODYMOD = 0
TRNQSMOD = 0 ACNQSMOD = 0 FNOIMOD = 1 + TNOIMOD = 0 DIOMOD = 1
PERMOD = 1 GEOMOD = 0
```

Fully scalable device

The fully scalable model has the same structure in principal as the single transistor model, but uses additional, scalable, external inductors and capacitors. It uses equations to determine the values for the substrate resistance parameters. These equations are derived from simple assumptions according to “MOS Transistor Modeling for RF IC Design” [3]. Please see the following schematic and cross section as well as the model equations for more details.

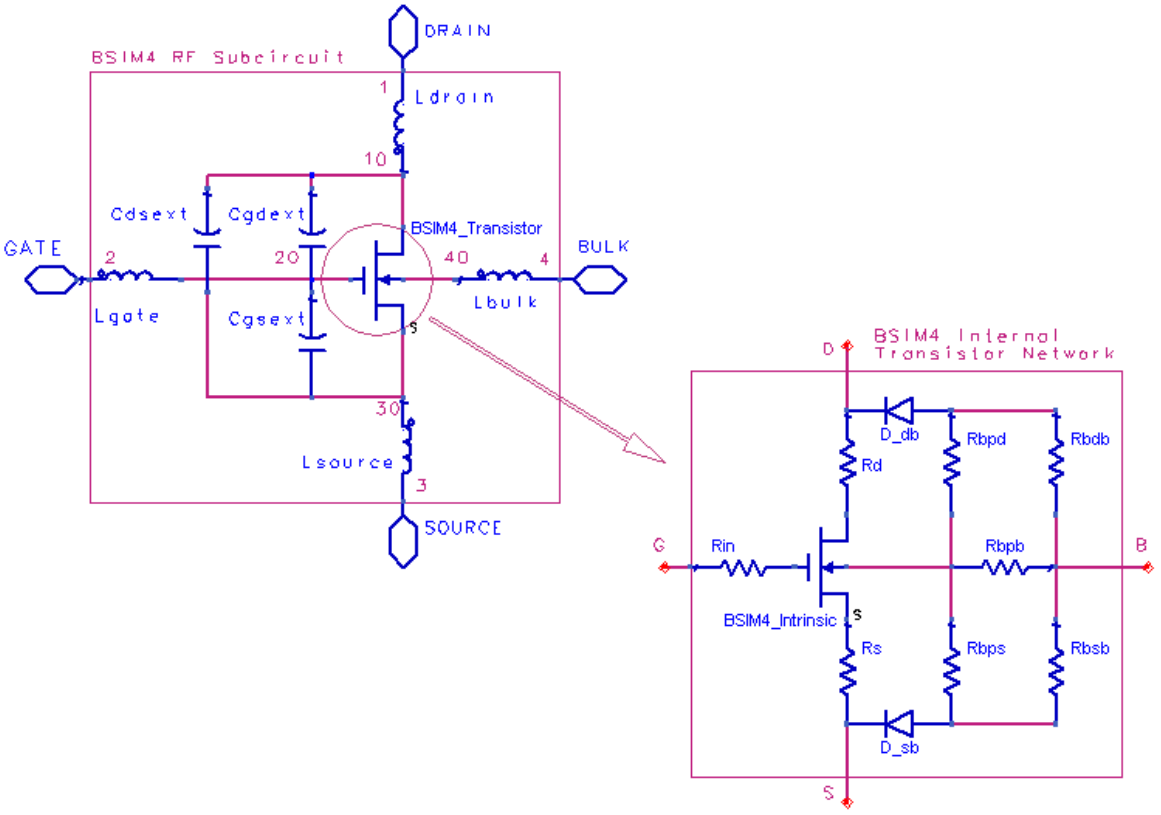


Figure 164 Scalable BSIM4 RF model

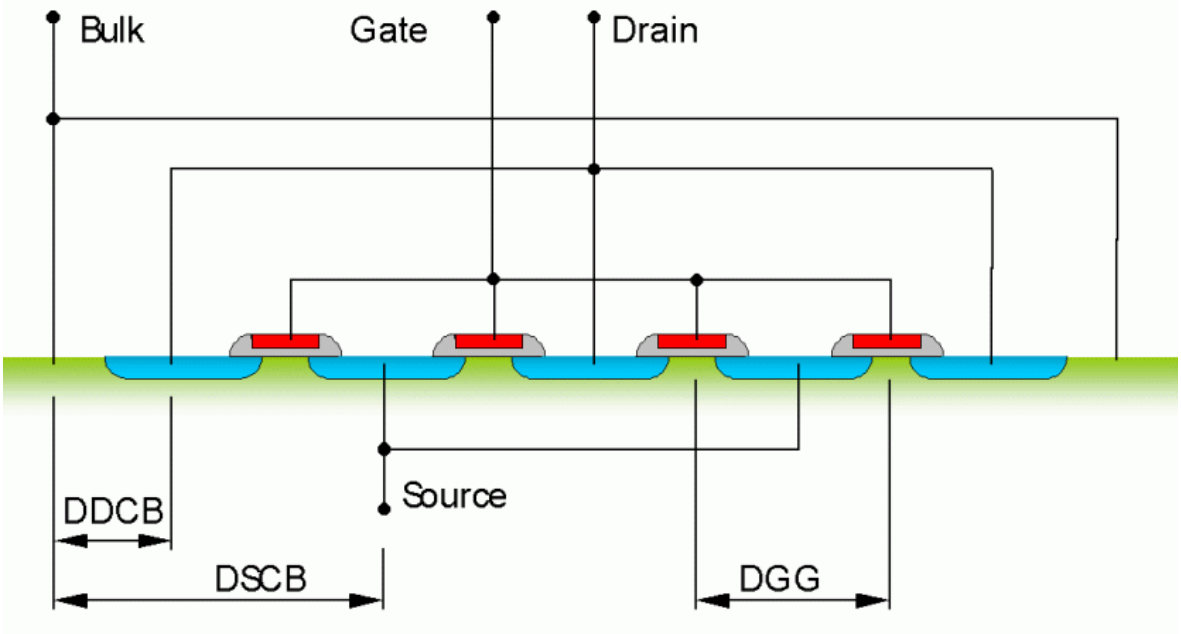


Figure 165 Cross Section of a Multifinger RF MOS Transistor

The substrate resistance network parameters RBPB, RBPD, RBPS, RBDB, and RBSB are derived using 4 new model parameters:

- RSHB: sheet resistance of the substrate
- DSCB: distance between the source contact and the outer source area
- DDCB: distance between the drain contact and the outer drain area
- DGG: distance between two gate stripes

Relevant Model Equations for Substrate Resistance Parameters

$$factor-even-odd = \frac{1}{2}\cdot\left(1+\left(NF-2\cdot int\left(\frac{NF}{2}\right)\right)\right)$$

$$RBPB = 1\times10^{9}$$

$$RBPD = RBPS = \frac{RSHB \cdot (L + DGG)}{2W}$$

$$RBSB = \frac{(factor - even - odd) \cdot NF \cdot DSCB \cdot RSHB}{W}$$

$$RBSD = \frac{(factor - even - odd) \cdot NF \cdot DDCB \cdot RSHB}{W}$$

Substrate contact resistance scaling

Due to different layouts of RF multifinger MOS transistors, BSIM4.3.0 has been enhanced with a new flag to model the substrate resistance according to the substrate contacts used. The flag, RSUB_EQ, can have two values:

- RSUB_EQ = 0 : symmetric substrate contacts
- RSUB_EQ = 1 : horseshoe substrate contact

For the horseshoe contact, a new dimension has been defined, as can be seen in Figure 169:

- DHSDBC: distance between Drain or Source edge and substrate contact of the horseshoe

Inside the Modeling Package, temporary parameters are used to calculate the values for the five resistances used inside BSIM4. Figure 166 as well as Figure 168 are showing the use of those temporary parameters, tmp_rdb1, tmp_rdb2, tmp_rsb1 and tmp_rsb2.

The following Figure 166 shows a RF multifinger MOS transistor with symmetric substrate contacts (use RSUB_EQ = 0), whereas Figure 167 shows a 3-dimensional view of such a contact.

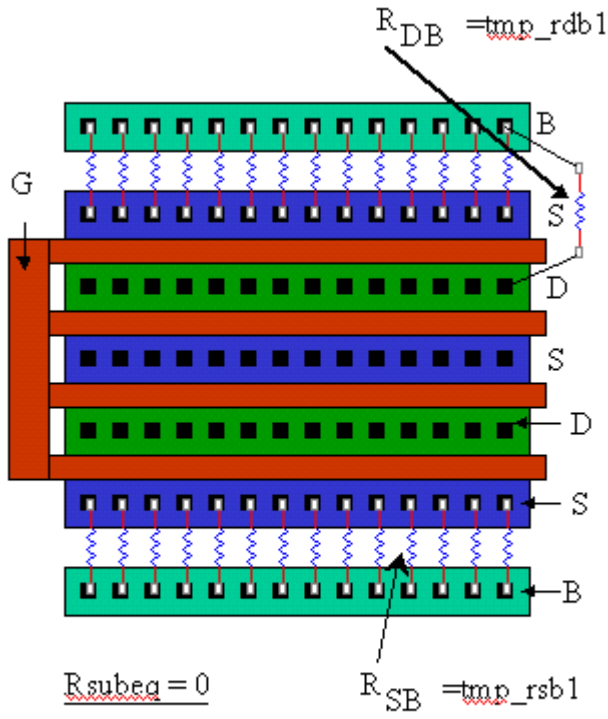


Figure 166 Symmetric substrate contacts, top view

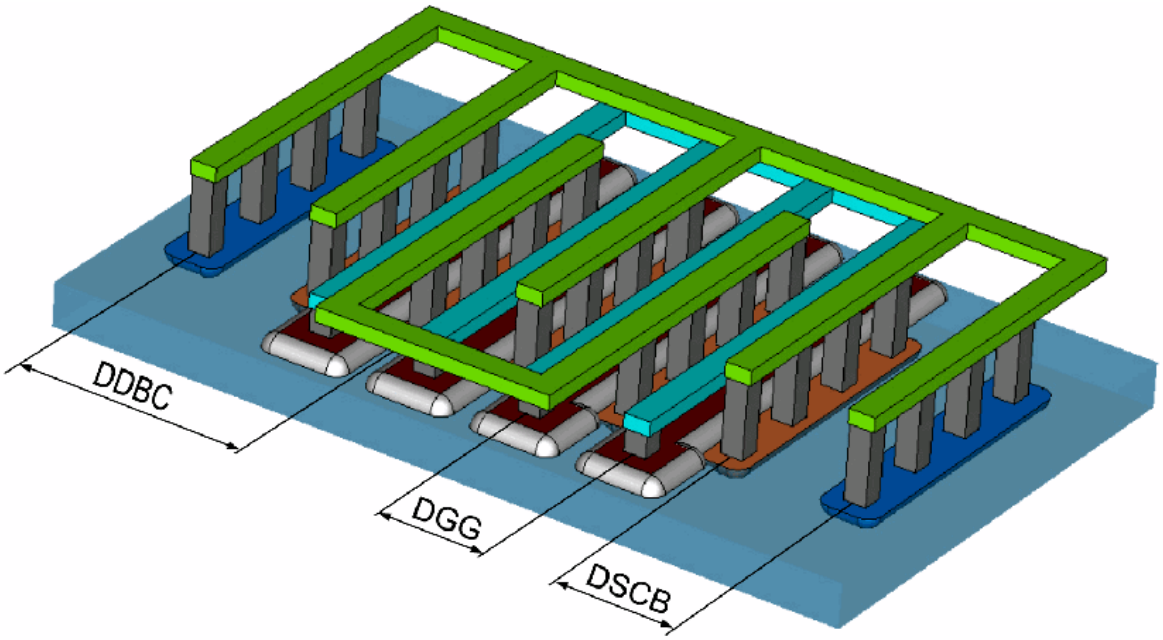


Figure 167 3-dimensional view of a symmetric substrate contact with Source connected to Substrate

The following equations are used to calculate substrate resistance temporary values inside the fully scalable model:

$$tmp\ldots rdb1 = \frac{(factor - even - odd) \cdot NF \cdot DDBC \cdot RSHB}{W}$$

$$tmp\ldots rsb1 = \frac{(factor - even - odd) \cdot NF \cdot DSCB \cdot RSHB}{W}$$

$$tmp\ldots rdb2 = \frac{DHSDBC \cdot RSHB}{NF \cdot (DGG + L)}$$

$$tmp\ldots rsb2 = \frac{DHSDBC \cdot RSHB}{NF \cdot (DGG + L)}$$

The following figures show the horseshoe substrate contact as well as a 3-dimensional view of such a contact (use RSUB_EQ=1).

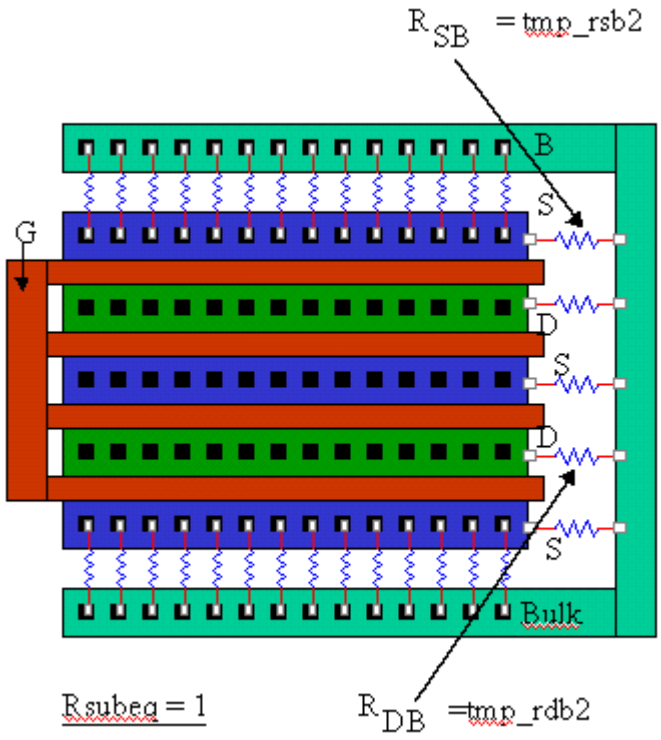


Figure 168 Horseshoe substrate contacts, top view

The temporary values calculated above are combined to give BSIM4 model parameters as follows:

RSUB_EQ = 0:

$$RBDB = RDB \cdot tmp\_rdb1$$

$$RBSB = RSB \cdot tmp\_rsb1$$

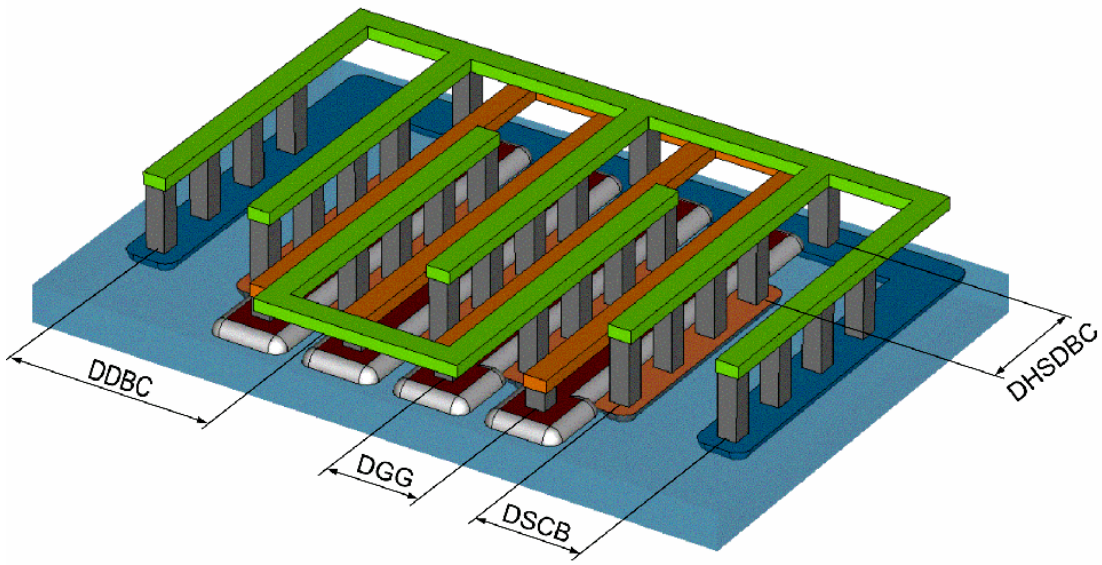


Figure 169 3-dimensional view of a horseshoe substrate contact with Source connected to Substrate

RSUB_EQ = 1:

$$RBDB = RDB \cdot \frac{(tmp...rdb1) \cdot (tmp...rdb2)}{(tmp...rdb1) + (tmp...rdb2)}$$

$$RBSB = RSB \cdot \frac{(tmp...rsb1) \cdot (tmp...rsb2)}{(tmp...rsb1) + (tmp...rsb2)}$$

In addition, the channel length variation inside a multifinger device is not constant [4]. This behavior is taken into account using a variable channel length variation as a function of the number of gate fingers. Using the parameters DL0, DL1, and DL2 this channel length variation can be set.

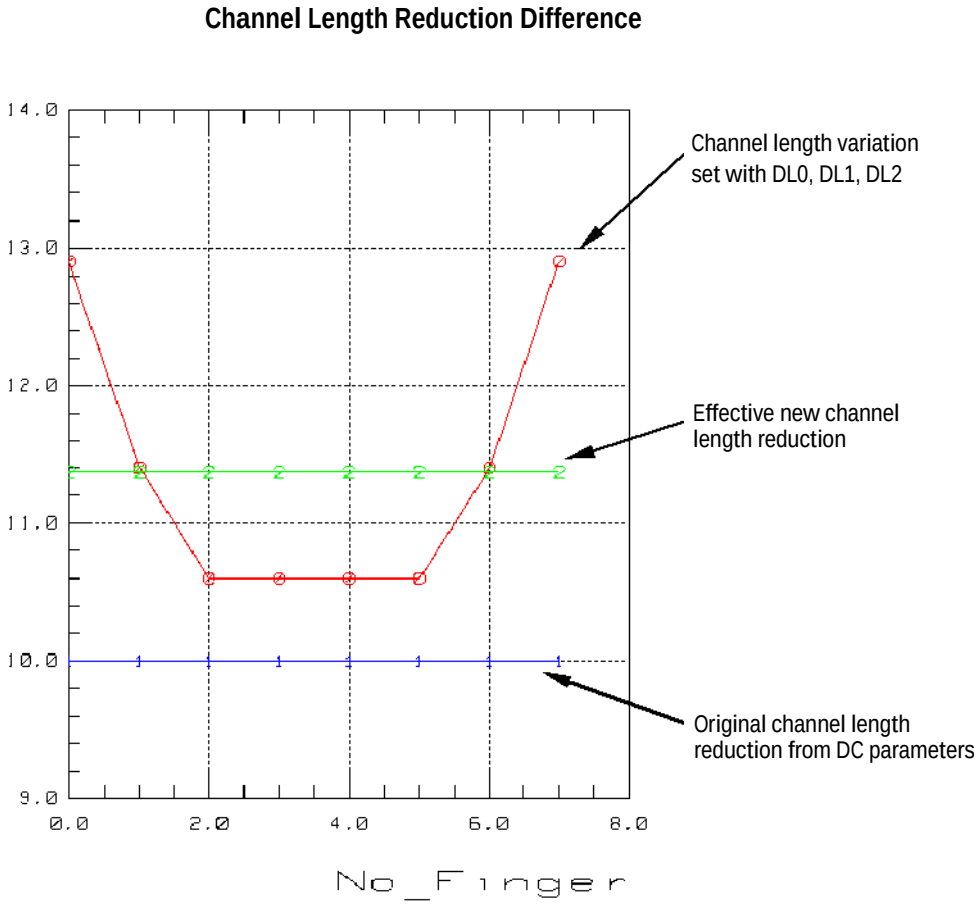


Figure 170 DL0, DL1, DL2 Channel Length Reduction Difference inside a Multifinger Device

The implementation of these enhancements requires the fully scalable model defined in a subcircuit. The following example shows the netlist implemented into the ADS simulator.

```
LINK CIRC “Circuit” { data { circuitdeck {
* ------------------------------------------------------------------------------
* Fully scalable subcircuit model for BSIM4.3.0 RF n-type devices * * Simulator: HSPICE
* Model: BSIM4 Modeling Package * Date: 14.09.2004
* Origin: ICCAP_ROOT/..../bsim4/circuits/hspice/cir/rf_nmos_scale.cir
*-------------------------------------------------------------------------------------*
*--- Information for model implementation ----------------------------------------------
* In hspice, call the sub circuit model as follows with the actual values of L, W, etc. *
* xrfmos 1 2 3 4 bsim4_rf_extract tmp_l = 0.25u tmp_w = 80u ..... *
* Please note, that according to the BSIM4 model definition, the parameters tmp_w, tmp_ad,
* tmp_as, tmp_pd, tmp_ps, tmp_nrs, tmp_nrd always define the TOTAL width (drain area, ...,
* number of drain squares) of the multi finger device.
* The width (drain area, ..) of a single finger of the multifinger MOSFET will be calculated
* inside the BSIM4 model using the instance parameter ‘NF’ and the selected ‘GEOMOD’
* parameter.
* .subckt bsim4_rf_extract 1 2 3 4
#echo + tmp_l=0.25e-6 tmp_w=10e-6 tmp_nf==1 tmp_ad=10e-12 tmp_as=10e-12 tmp_pd=22e-6
#echo + tmp_ps=22e-6 tmp_sa=0 tmp_sb=0 tmp_sd=0 tmp_nrd=0 tmp_nrs=0 tmp_geomod=0
#echo + tmp_rgeomod=0 tmp_min=0 *
*--- BSIM4 model card ------------------------------------------------------------------
#echo .MODEL BSIM4_HF NMOS #echo + LEVEL = $mpar(LEVEL = 54) VERSION = 4.30
#echo + BINUNIT = $mpar(BINUNIT=2) PARAMCHK = $mpar(PARAMCHK=1)
#echo + MOBMOD = $mpar(MOBMOD=1) RDSMOD = $mpar(RDSMOD=0) IGCMOD = $mpar(IGCMOD=0)
#echo + IGBMOD = $mpar(IGBMOD=0) CAPMOD = $mpar(CAPMOD=2) RGATEMOD = $mpar(RGATEMOD=2)
#echo + RBODYMOD = $mpar(RBODYMOD=1) TRNQSMOD = $mpar(TRNQSMOD=0)
#echo + ACNQSMOD = $mpar(ACNQSMOD=0) FNOIMOD = $mpar(FNOIMOD=1) TNOIMOD = $mpar(TNOIMOD=0)
#echo + DIOMOD = $mpar(DIOMOD=1) PERMOD = $mpar(PERMOD=1) EPSROX = $mpar(EPSROX=3.9)
#echo + TOXE = $mpar(TOXE=3E-9) TOXP = $mpar(TOXP=3E-9) TOXM = $mpar(TOXM=3E-9)
#echo + DTOX = $mpar(DTOX=0)
#echo + XJ = $mpar(XJ=1.5E-7) NDEP = $mpar(NDEP=1.7E17) NGATE = $mpar(NGATE=0)
#echo + NSD = $mpar(NSD=1E20) XT = $mpar(XT=1.55E-7) RSH = $mpar(RSH=0)
#echo + RSHG = $mpar(RSHG=0.1)
#echo + VTH0 = $mpar(VTH0=0.7) PHIN = $mpar(PHIN=0) K1 = $mpar(K1=0.33)
#echo + K2 = $mpar(K2=-0.018) K3 = $mpar(K3=2) K3B = $mpar(K3B=0) W0 = $mpar(W0=2.5E-6)
#echo + LPE0 = $mpar(LPE0=1.74E-7) LPEB = $mpar(LPEB=0) VBM = $mpar(VBM=-3)
#echo + DVT0 = $mpar(DVT0=2.2) DVT1 = $mpar(DVT1=0.53) DVT2 = $mpar(DVT2=-0.032)
#echo + DVTP0 = $mpar(DVTP0=0) DVTP1 = $mpar(DVTP1=0.001) DVT0W = $mpar(DVT0W=0)
#echo + DVT1W = $mpar(DVT1W=5.3E6) DVT2W = $mpar(DVT2W=-0.032) ETA0 = $mpar(ETA0=0.08)
#echo + ETAB = $mpar(ETAB=-0.07)
#echo + DSUB = $mpar(DSUB=0.56) U0 = $mpar(U0=0.067) UA = $mpar(UA=1E-9)
#echo + UB = $mpar(UB=1E-19) UC = $mpar(UC=-0.0465)
#echo + EU = $mpar(EU=1.67) VSAT = $mpar(VSAT=8E4) A0 = $mpar(A0=1) AGS = $mpar(AGS=0)
#echo + B0 = $mpar(B0=0) B1 = $mpar(B1=0) KETA = $mpar(KETA=-0.047) A1 = $mpar(A1=0)
#echo + A2 = $mpar(A2=1) VOFF = $mpar(VOFF=-0.08) VOFFL = $mpar(VOFFL=0)
#echo + MINV = $mpar(MINV=0) NFACTOR = $mpar(NFACTOR=1) CIT = $mpar(CIT=0)
#echo + CDSC = $mpar(CDSC=2.4E-4) CDSCB = $mpar(CDSCB=0) CDSCD = $mpar(CDSCD=0)
#echo + PCLM = $mpar(PCLM=1.3) PDIBLC1 = $mpar(PDIBLC1=0.39) PDIBLC2 = $mpar(PDIBLC2=0.0086)
#echo + PDIBLCB = $mpar(PDIBLCB=0) DROUT = $mpar(DROUT=0.56) PSCBE1 = $mpar(PSCBE1=4.24E8)
#echo + PSCBE2 = $mpar(PSCBE2=1E-5) PVAG = $mpar(PVAG=0)
#echo + DELTA = $mpar(DELTA=0.01) FPROUT = $mpar(FPROUT=0) PDITS = $mpar(PDITS=1M)
#echo + PDITSL = $mpar(PDITSL=0) PDITSD = $mpar(PDITSD=0)
#echo + RDSW = $mpar(RDSW=200) RDSWMIN = $mpar(RDSWMIN=0) RDW = $mpar(RDW=100)
#echo + RDWMIN = $mpar(RDWMIN=0) RSW = $mpar(RSW=100)
```

```
#echo + RSWMIN = $mpar(RSWMIN=0) PRWG = $mpar(PRWG=1) PRWB = $mpar(PRWB=0) WR = $mpar(WR=1)
#echo + LINT = $mpar(LINT=0) WINT = $mpar(WINT=0) DWG = $mpar(DWG=0) DWB = $mpar(DWB=0)
#echo + WL = $mpar(WL=0) WLN = $mpar(WLN=1) WW = $mpar(WW=0) WWN = $mpar(WWN=1)
#echo + WWL = $mpar(WWL=0) LL = $mpar(LL=0) LLN = $mpar(LLN=1) LW = $mpar(LW=0)
#echo + LWN = $mpar(LWN=1) LWL = $mpar(LWL=0) LLC = $mpar(LLC=0) LWC = $mpar(LWC=0)
#echo + LWLC = $mpar(LWLC=0) WLC = $mpar(WLC=0) WWC = $mpar(WWC=0)
#echo + ALPHA0 = $mpar(ALPHA0=1E-5) ALPHA1 = $mpar(ALPHA1=0) BETA0 = $mpar(BETA0=15)
#echo + AGIDL = $mpar(AGIDL=0) BGIDL = $mpar(BGIDL=2.3E9) CGIDL = $mpar(CGIDL=0.5)
#echo + EGIDL = $mpar(EGIDL=0.8)
#echo + AIGBACC = $mpar(AIGBACC=0.43) BIGBACC = $mpar(BIGBACC=0.054)
#echo + CIGBACC = $mpar(CIGBACC=0.075) NIGBACC = $mpar(NIGBACC=1)
#echo + AIGBINV = $mpar(AIGBINV=0.35) BIGBINV = $mpar(BIGBINV=0.03)
#echo + CIGBINV = $mpar(CIGBINV=0.006) EIGBINV = $mpar(EIGBINV=1.1)
#echo + NIGBINV = $mpar(NIGBINV=3) AIGC = $mpar(AIGC=0.54) BIGC = $mpar(BIGC=0.054)
#echo + CIGC = $mpar(CIGC=0.075) AIGSD = $mpar(AIGSD=0.43) BIGSD = $mpar(BIGSD=0.054)
#echo + CIGSD = $mpar(CIGSD=0.075) DLCIG = $mpar(DLCIG=0) NIGC = $mpar(NIGC=1)
#echo + POXEDGE = $mpar(POXEDGE=1) PIGCD = $mpar(PIGCD=1) NTOX = $mpar(NTOX=1)
#echo + TOXREF = $mpar(TOXREF=3E-9)XPART = $mpar(XPART=0) CGSO = $mpar(CGSO=0)
#echo + CGDO = $mpar(CGDO=0) CGBO = $mpar(CGBO=0) CGSL = $mpar(CGSL=0)
#echo + CGDL = $mpar(CGDL=0) CKAPPAS = $mpar(CKAPPAS=0.6) CKAPPAD = $mpar(CKAPPAD=0.6)
#echo + CF = $mpar(CF=0) CLC = $mpar(CLC=1E-7) CLE = $mpar(CLE=0.6) DLC = $mpar(DLC=0)
#echo + DWC = $mpar(DWC=0) VFBCV = $mpar(VFBCV=-1) NOFF = $mpar(NOFF=1)
#echo + VOFFCV = $mpar(VOFFCV=0) ACDE = $mpar(ACDE=1) MOIN = $mpar(MOIN=15)
#echo + XRCRG1 = $mpar(XRCRG1=12) XRCRG2 = $mpar(XRCRG2=1) RBPB = $mpar(RBPB=50)
#echo + RBPD = $mpar(RBPD=50) RBPS = $mpar(RBPS=15) RBDB = $mpar(RBDB=50)
#echo + RBSB = $mpar(RBSB=50) GBMIN = $mpar(GBMIN=1E-12)
#echo + NOIA = $mpar(NOIA=6.25E41) NOIB = $mpar(NOIB=3.125E26) NOIC = $mpar(NOIC=8.75)
#echo + EM = $mpar(EM=4.1E7) AF = $mpar(AF=1) EF = $mpar(EF=1) KF = $mpar(KF=0)
#echo + NTNOI = $mpar(NTNOI=1) TNOIA = $mpar(TNOIA=1.5) TNOIB = $mpar(TNOIB=3.5)
#echo + DMCG = $mpar(DMCG=0) DMCI = $mpar(DMCI=0) DMDG = $mpar(DMDG=0)
#echo + DMCGT = $mpar(DMCGT=0) DWJ = $mpar(DWJ=0) XGW = $mpar(XGW=0) XGL = $mpar(XGL=0)
#echo + NGCON = $mpar(NGCON=1) XL = $mpar(XL=0) XW = $mpar(XW=0)
#echo + IJTHSREV = $mpar(IJTHSREV=0.1) IJTHSFWD = $mpar(IJTHSFWD=0.1) XJBVS = $mpar(XJBVS=1)
#echo + BVS = $mpar(BVS=10) JSS = $mpar(JSS=1E-4) JSWS = $mpar(JSWS=0)
#echo + JSWGS = $mpar(JSWGS=0) CJS = $mpar(CJS=5E-4) MJS = $mpar(MJS=0.5)
#echo + MJSWS = $mpar(MJSWS=0.33) #echo + CJSWS = $mpar(CJSWS=5E-10)
#echo + CJSWGS = $mpar(CJSWGS=5E-10) MJSWGS = $mpar(MJSWGS=0.33) PBS = $mpar(PBS=1)
#echo + PBSWS = $mpar(PBSWS=1) PBSWGS = $mpar(PBSWGS=1)
#echo + IJTHDREV = $mpar(IJTHDREV=0.1) IJTHDFWD = $mpar(IJTHDFWD=0.1) XJBVD = $mpar(XJBVD=1)
#echo + BVD = $mpar(BVD=10) JSD = $mpar(JSD=1E-4) JSWD = $mpar(JSWD=0)
#echo + JSWGD = $mpar(JSWGD=0) CJD = $mpar(CJD=5E-4) MJD = $mpar(MJD=0.5)
#echo + MJSWD = $mpar(MJSWD=0.33) CJSWD = $mpar(CJSWD=5E-10) CJSWGD = $mpar(CJSWGD=5E-10)
#echo + MJSWGD = $mpar(MJSWGD=0.33) PBD = $mpar(PBD=1) PBSWD = $mpar(PBSWD=1)
#echo + PBSWGD = $mpar(PBSWGD=1)
#echo + SAREF = $mpar(SAREF=1E-6) SBREF = $mpar(SBREF=1E-6) WLOD = $mpar(WLOD=2E-6)
#echo + KU0 = $mpar(KU0=4E-6) KVSAT = $mpar(KVSAT=0.2) KVTH0 = $mpar(KVTH0=-2E-8)
#echo + TKU0 = $mpar(TKU0=0.0) LLODKU0 = $mpar(LLODKU0=1.1) WLODKU0 = $mpar(WLODKU0=1.1)
#echo + LLODVTH = $mpar(LLODVTH=1.0) #echo + WLODVTH = $mpar(WLODVTH=1.0)
#echo + LKU0 = $mpar(LKU0=1E-6) WKU0 = $mpar(WKU0=1E-6) PKU0 = $mpar(PKU0=0.0)
#echo + LKVTH0 = $mpar(LKVTH0=1.1E-6) WKVTH0 = $mpar(WKVTH0=1.1E-6)
#echo + PKVTH0 = $mpar(PKVTH0=0.0) STK2 = $mpar(STK2=0.0) LODK2 = $mpar(LODK2=1.0)
#echo + STETA0 = $mpar(STETA0=0.0) LODETA0 = $mpar(LODETA0=1.0) LAMBDA = $mpar(LAMBDA=0)
#echo + VTL = $mpar(VTL=2e5) LC = $mpar(LC=0) XN = $mpar(XN=3)
```

```
#echo + TEMPMOD = $mpar(TEMPMOD=0)
#echo + TNOM = $mpar(TNOM=27) UTE = $mpar(UTE=-1.5) KT1 = $mpar(KT1=-0.11)
#echo + KT1L = $mpar(KT1L=0) KT2 = $mpar(KT2=0.022)
#echo + UA1 = $mpar(UA1=1E-9) UB1 = $mpar(UB1=-1E-18) UC1 = $mpar(UC1=0.067)
#echo + AT = $mpar(AT=3.3E4) PRT = $mpar(PRT=0) NJS = $mpar(NJS=1) NJD = $mpar(NJD=1)
#echo + XTIS = $mpar(XTIS=3) XTID = $mpar(XTID=3) TPB = $mpar(TPB=0)
#echo + TPBSW = $mpar(TPBSW=0) TPBSWG = $mpar(TPBSWG=0) TCJ = $mpar(TCJ=0)
#echo + TCJSW = $mpar(TCJSW=0) TCJSWG = $mpar(TCJSWG=0)
#echo + *
*--- Extension to BSIM4 to enable:
* - scalable external capacitors and inductors to account for cross coupling in
* the metal stripes and additional delay due to large sizes
* - a scalable substrate network using different configurations (symmetric / horseshoe)
* - scalable Delta L reduction *
* Parameters: *
* CGDEXT0 external capacitance gate - drain per gate width and gate finger [F/m]
* CGSEXT0 external capacitance gate - source per gate width and gate finger [F/m]
* CDSEXT0 external capacitance drain - source per gate width and gate finger [F/m]
* LDRAIN0 drain inductance per gate width and gate finger [H/m]
* LGATE0 gate inductance per gate width and gate finger [H/m]
* LSOURCE0 source inductance per gate width and gate finger [H/m]
* LBULK0 bulk inductance per gate width and gate finger [H/m]
* RSHB bulk sheet resistance [Ohm sq]
* DSBC distance source implant to bulk contact [m]
* DDBC distance drain implant to bulk contact [m]
* DHSDBC distance drain/source edge to horseshoe substrate contact [m]
* DGG distance gate to gate [m]
* DL0 basic channel length reduction correction [m]
* DL1 channel length reduction correction 1. and 2. outer fingers [m]
* DL2 :
* RSUBEQ selection flag for different substrate resistance configurations [-]
* RSUBEQ=0: symmetric substrate resistance contacts
* RSUBEQ=1: horseshoe substrate resistance contacts *
#echo .PARAM CGDEXT0 = $mpar(CGDEXT0=1e-9) .param CGSEXT0 = $mpar(CGSEXT0=1e-9)
#echo .param CDSEXT0 = $mpar(CDSEXT0=1e-9) .param LDRAIN0 = $mpar(LDRAIN0=1e-6)
#echo .param LGATE0 = $mpar(LGATE0=1e-6) .param LSOURCE0 = $mpar(LSOURCE0=1e-6)
#echo .param LBULK0 = $mpar(LBULK0=1e-6) .param RSHB = $mpar(RSHB=25)
#echo .param DSBC = $mpar(DSBC=2e-6) .param DDBC = $mpar(DDBC=2e-6)
#echo .param DHSDBC = $mpar(DHSDBC=2e-6) .param DGG = $mpar(DGG=2e-6)
#echo .param DL0 = $mpar(DL0=0) .param DL1 = $mpar(DL1=0) .param DL2 = $mpar(DL2=0)
#echo .param RSUBEQ = $mpar(RSUBEQ=0) * *
*--- temporary constants ---------------------
#echo .param factor_even_odd = '0.5*(1+(tmp_nf-2*int(0.5*tmp_nf)))'
#echo .param tmp_dl1 = '(tmp_nf-4.5)/(2*abs(tmp_nf-4.5)) * 8/ tmp_nf'
#echo .param tmp_dl2 = '(tmp_nf-2.5)/(2*abs(tmp_nf-2.5)) * 4/ tmp_nf'
;--- calculation of substrate resistance for different configurations
#echo .param tmp_rdb1 = 'factor_even_odd*tmp_nf*DDBC*RSHB / tmp_w'
#echo .param tmp_rsb1 = 'factor_even_odd*tmp_nf*DSBC*RSHB / tmp_w'
#echo .param tmp_rdb2 = 'DHSDBC*RSHB / (tmp_nf*(DGG+tmp_l))'
#echo .param tmp_rsb2 = 'tmp_rdb2'
* RSUBEQ=0 symmetric substrate contacts
#echo .param tmp_rdb_rsubeq0 = 'tmp_rdb1'
#echo .param tmp_rsb_rsubeq0 = 'tmp_rsb1'
```

```
* RSUBEQ=1 horseshoe substrate contacts
#echo .param tmp_rdb_rsubeq1 = '(tmp_rdb1*tmp_rdb2)/(tmp_rdb1+tmp_rdb2)'
#echo .param tmp_rsb_rsubeq1 = '(tmp_rsb1*tmp_rsb2)/(tmp_rsb1+tmp_rsb2)'
* flag to select substrate equations
#echo .param tmp_flag_rsubeq0 = '1/(1+abs(RSUBEQ)*1e9)'
#echo .param tmp_flag_rsubeq1 = '1/(1+abs(RSUBEQ-1)*1e9)'
#echo .param tmp_rbdb = 'tmp_flag_rsubeq0*tmp_rdb_rsubeq0 tmp_flag_rsubeq1*tmp_rdb_rsubeq1'
#echo .param tmp_rbsb = 'tmp_flag_rsubeq0*tmp_rsb_rsubeq0 tmp_flag_rsubeq1*tmp_rsb_rsubeq1'
*
* --------- Gate network -----------------------------
#echo CGDEXT 20 10 'CGDEXT0*tmp_w'
#echo CGSEXT 20 30 'CGSEXT0*tmp_w'
#echo LGATE 2 20 'LGATE0*tmp_w' *
* --------- Drain network ----------------------------
#echo LDRAIN 1 10 'LDRAIN0*tmp_w'
#echo CDSEXT 10 30 'CDSEXT0*tmp_w' *
* --------- Source network ---------------------------
#echo LSOURCE 3 30 'LSOURCE0*tmp_w' *
* --------- Substrate network ------------------------
#echo LBULK 4 40 'LBULK0*tmp_w' *
*--- call fully scalable MOSFET ---------------------------------------------------------
#echo M1 10 20 30 40 BSIM4_HF
#echo + L='tmp_l - 2*(DL0+tmp_dl1*DL1+tmp_dl2*DL2)' W=tmp_w NF=tmp_nf
#echo + AD=tmp_ad AS=tmp_as PD=tmp_pd PS=tmp_ps SA=tmp_sa SB=tmp_sb SD=tmp_sd
#echo + NRD=tmp_nrd NRS=tmp_nrs
#echo + GEOMOD=tmp_geomod RGEOMOD=tmp_rgeomod MIN=tmp_min RBPB=1e9
#echo + RBPS='0.5*RSHB*(tmp_l+DGG) / tmp_w'
#echo + RBPD='0.5*RSHB*(tmp_l+DGG) / tmp_w'
#echo + RBSB='tmp_rbsb'
#echo + RBDB='tmp_rbdb' * .ends } } }
```

Test Structures for Deep Submicron CMOS Processes

A very important prerequisite for a proper model parameter extraction is the selection of appropriate test structures.

Chapter 4, “BSIM3v3 Characterization” contains detailed descriptions of appropriate test structures for deep submicron MOS transistors. See Table 20 on page 296 for an example.

A very detailed description of ideal test structures are in the JESSI AC-41 reports [2].

Tutorials for the BSIM4 Modeling Package

This section provides some background information on what's behind the extraction routines used inside the BSIM4 Modeling Package. It will give you the ability to see what happens during an extraction step. Selecting one of the predefined extraction routines opens diagrams of relevant device behavior curves. The diagrams consist of measured data shown together with simulated curves using default parameter values provided. Windows open that show sliders for the parameters to be extracted during this extraction step. Now you are able to change the default parameters using the sliders. Once you have changed a parameter, a simulation starts using the newly entered or changed parameter. This gives you the opportunity to study parameter influence on device behavior. You can see which part of the diagrams are sensitive to the changed parameter.

Opening the Tutorial

Information

On opening the tutorial model file, you get a screen similar to the following one:

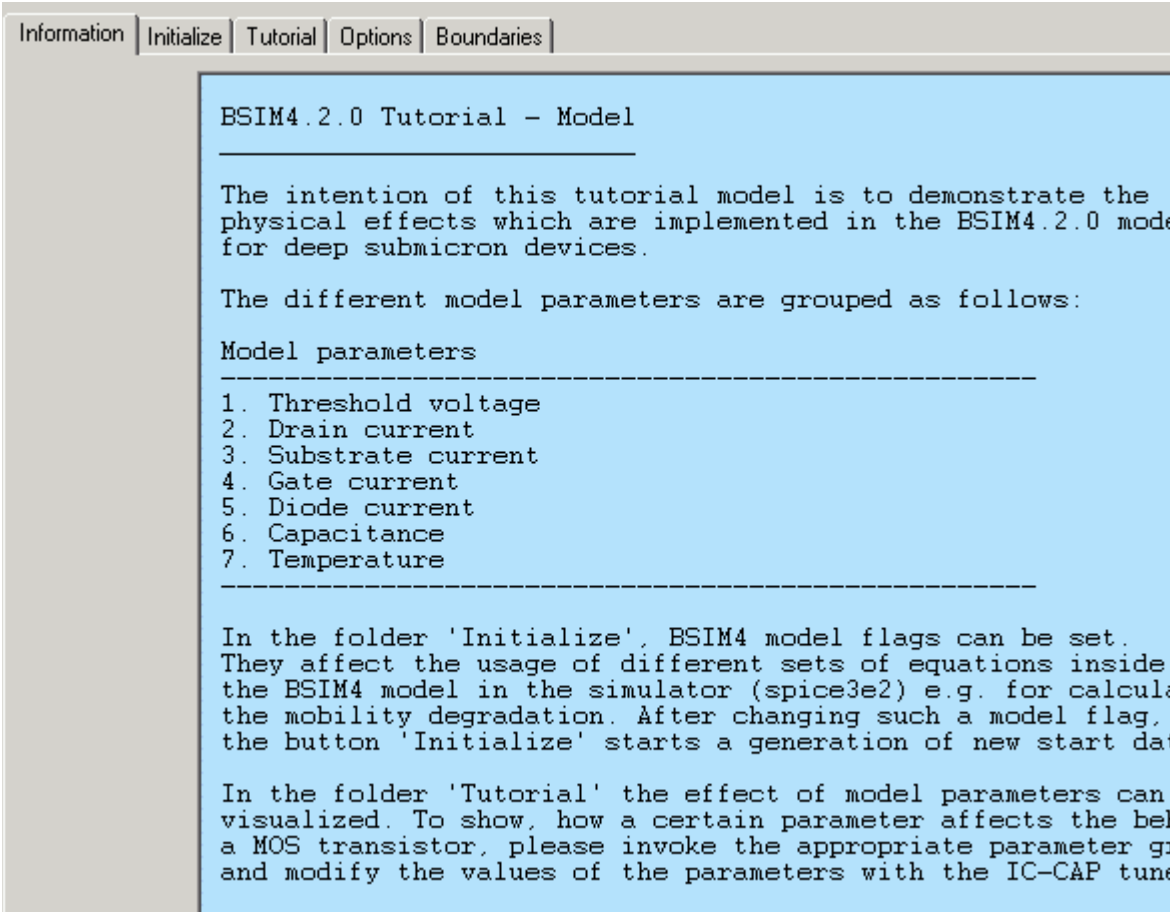
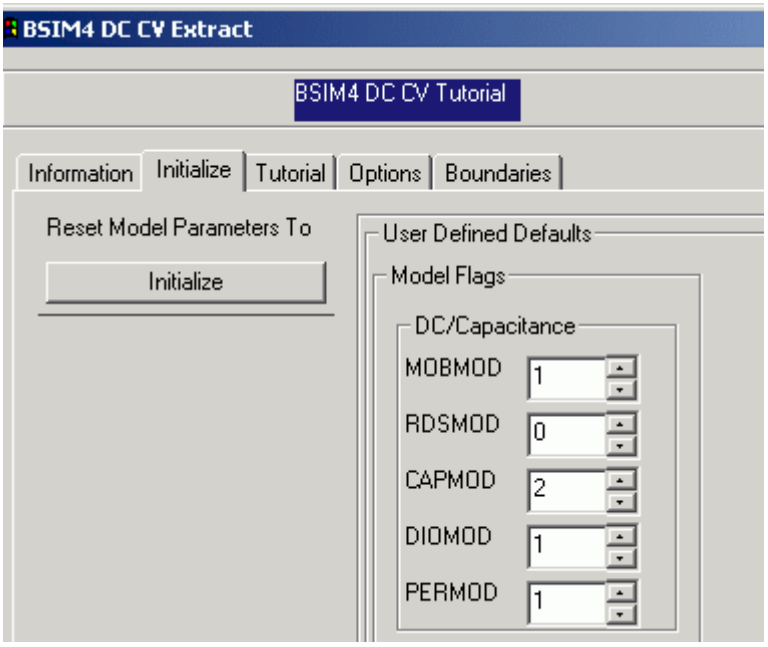

Information
Initialize
Tutorial
Options
Boundaries
BSIM4.2.0 Tutorial - Model
The intention of this tutorial model is to demonstrate the
physical effects which are implemented in the BSIM4.2.0 mode
for deep submicron devices.
The different model parameters are grouped as follows:
Model parameters
1. Threshold voltage
2. Drain current
3. Substrate current
4. Gate current
5. Diode current
6. Capacitance
7. Temperature
In the folder 'Initialize', BSIM4 model flags can be set.
They affect the usage of different sets of equations inside
the BSIM4 model in the simulator (spice3e2) e.g. for calcula
the mobility degradation. After changing such a model flag,
the button 'Initialize' starts a generation of new start da
In the folder 'Tutorial' the effect of model parameters can
visualized. To show, how a certain parameter affects the bel
a MOS transistor, please invoke the appropriate parameter gr
and modify the values of the parameters with the IC-CAP tune

Figure 171 Information screen of the BSIM4 tutorial model

Inside the Information folder, you get the information necessary to start with the tutorial.

Initialize

The next folder to the right, Initialize, is used to set initial values of BSIM4 flags used for simulation.



Tutorial

On entering the Tutorial folder, you will see the parameters extracted from different device curves, grouped for their influence on the device behavior. The following figure shows the Tutorial folder.

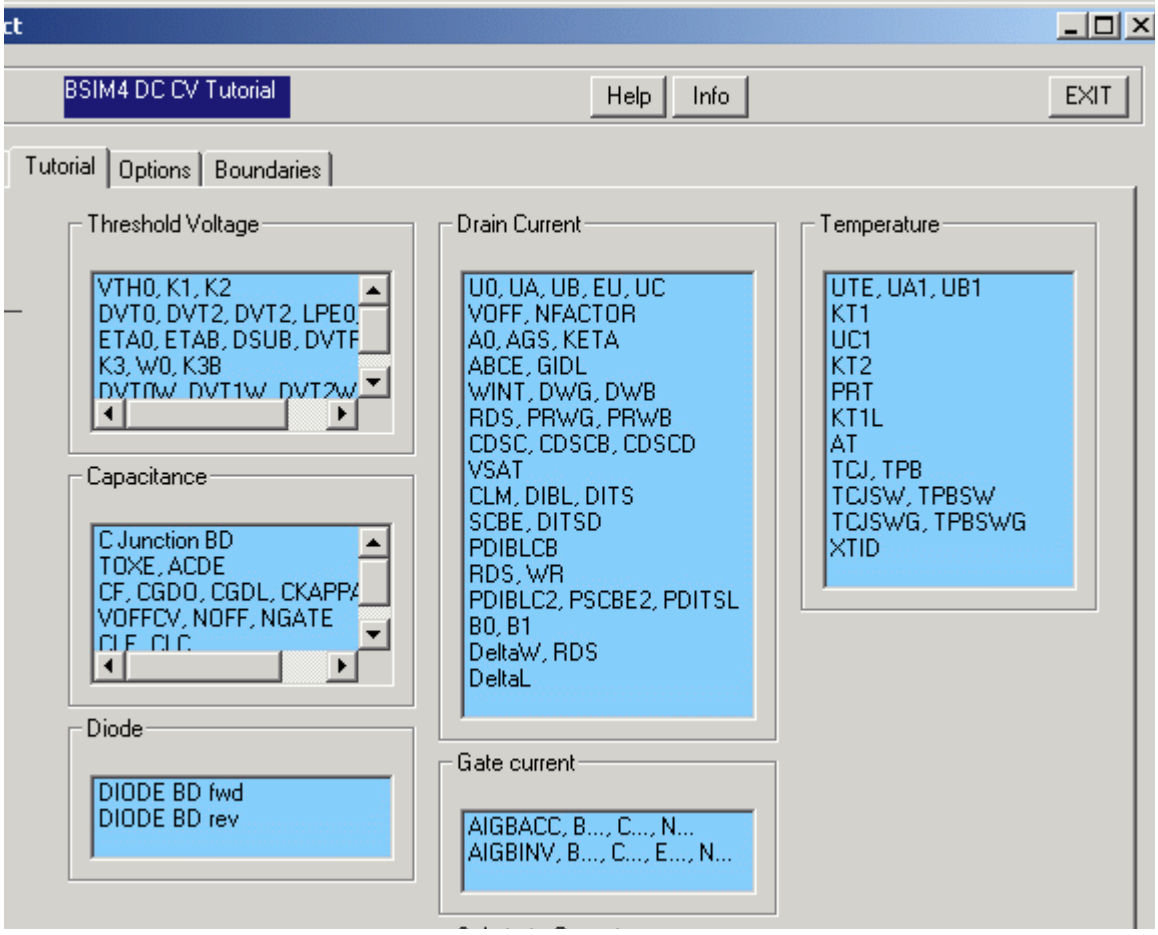


Figure 172 Tutorial folder

Selecting a line of parameters inside one of the different groups opens simulated device diagrams together with the IC-CAP tuner (see Figure 173).

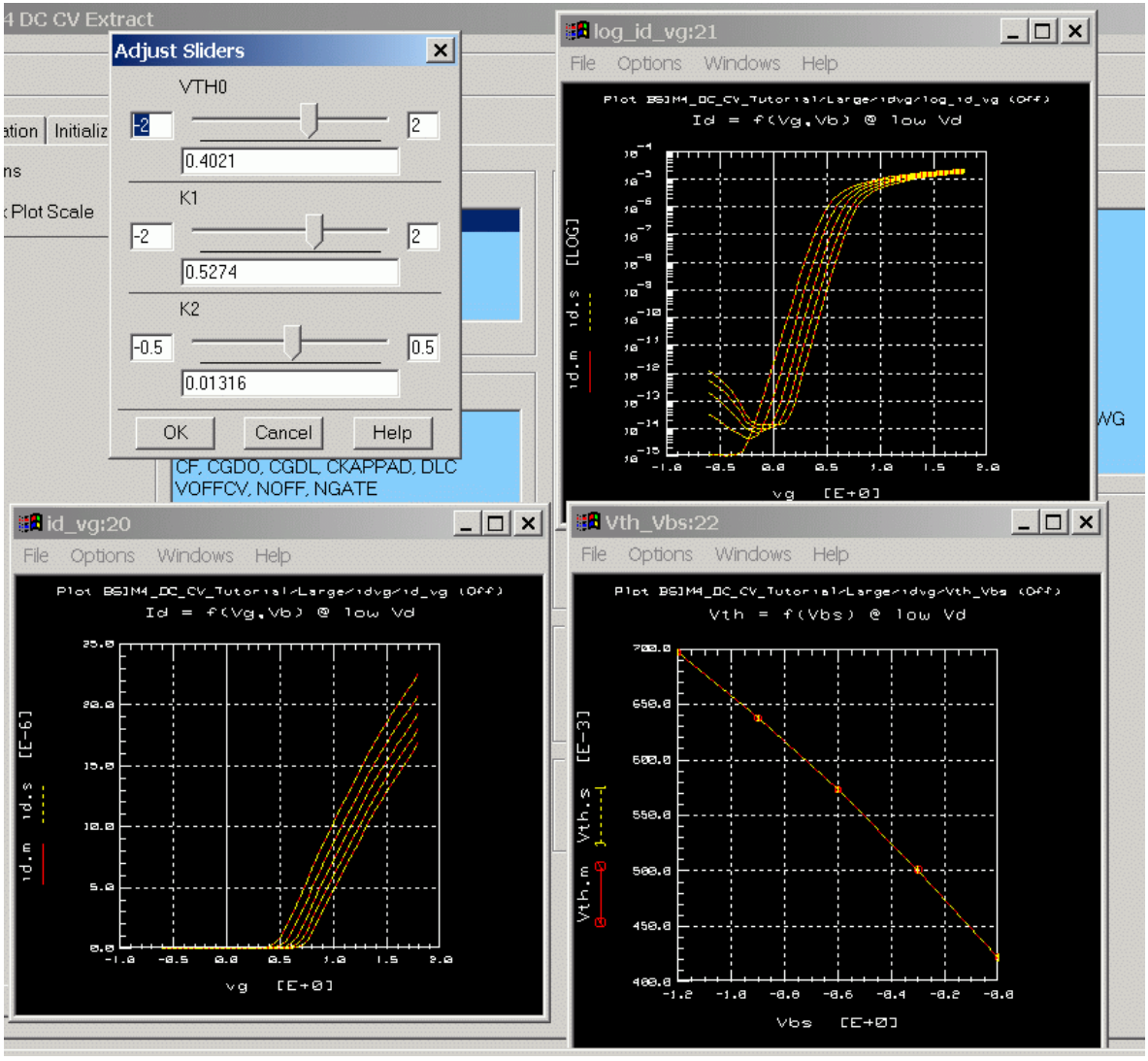


Figure 173 Diagrams opened for the Tutorial

In the example above, the line reading *VTH0, K1, K2* inside the *Threshold Voltage* group of parameters has been selected. The diagrams show simulated (yellow color) together with a set of measured (red color) parameters. If

you change the setting of one of the sliders inside the tuner window, the new value is taken for a simulation of the device behavior in that specific diagram and you can easily see the influence of the named parameter.

Options

Inside the Options folder, you can define the size and background of the plots to be displayed.

Boundaries

The Boundaries folder is described already in detail in "BSIM4 — Boundaries" on page 174.

SPICE Model Parameters

There are a number of new model parameters introduced with BSIM4.3.0, mainly associated with the newly introduced stress effect. Since the user of the former model revision, BSIM4.2.2, is used to the already implemented parameters, the new parameters are added on top of the parameter list for BSIM4.

The model parameters of the BSIM4 model can be divided into several groups. The main model parameters are used to model the key physical effects in the DC and CV behavior of submicron MOS devices at room temperature. Here they are grouped into subsections related to the physical effects of the MOS transistor. The second group of parameters are the process related parameters. They should only be changed if a detailed knowledge of a certain MOS production process is given. The third group of parameters are the temperature modeling parameters. The following two groups are used to model the AC and noise behavior of the MOS transistor. Finally the last group contains flags to select certain modes of operations and user definable model parameters. For more details about these operation modes, refer to the BSIM4 manual [1].

New Model Parameters for BSIM4.3.0

Table 39 New Parameters introduced with BSIM4.3.0

Parameter	Description	Default Value	Unit
	Stress Effect related Parameters		
SA	Instance parameter: Distance between OD edge to poly Si from one side, see Figure 161 If not given or ≤ 0, stress effect will be turned off!	0.0	m
SB	Instance parameter: Distance between OD edge to poly Si from the other side, see Figure 161 If not given or ≤ 0, stress effect will be turned off!	0.0	m

Table 39 New Parameters introduced with BSIM4.3.0 (continued)

Parameter	Description	Default Value	Unit
SD	Instance parameter: Distance between neighboring fingers, see Figure 161 For NF > 1: if not given or ≤ 0, stress effect will be turned off!	0.0	m
SAREF	Reference distance between OD edge to poly Si from one side	1E-6	m
SBREF	Reference distance between OD edge to poly Si from the other side	1E-6	m
WLOD	Width parameter for stress effect	0.0	m
KU0	stress effect mobility degradation/enhancement coefficient	0.0	1/m
KVSAT	Stress effect saturation velocity degradation/enhancement parameter $-1 \leq KVSAT \leq 1$	0.0	
TKU0	KU0 temperature coefficient	0.0	
LKU0	KU0 length dependence	0.0	
WKU0	KU0 width dependence	0.0	
PKU0	KU0 cross-term dependence	0.0	
LLODKU0	Length parameter for U0 stress effect (>0)	0.0	
WLODKU0	width parameter for U0 stress effect (>0)	0.0	
KVTH0	stress effect threshold shift parameter	0.0	
LKVTH0	KVTH0 length dependence	0.0	
WKVTH0	KVTH0 width dependence	0.0	
PKVTH0	KVTH0 cross-term dependence	0.0	
LLODVTH	VTH stress effect length parameter (>0)	0.0	
WLODVTH	VTH stress effect width parameter (>0)	0.0	
STK2	Shift factor for K2 with changing VTH0	0.0	
LODK2	K2 shift modification factor for stress effect (>0)	1.0	
STETA0	Shift factor for ETA0, related to change of VTH0	0.0	
LODETA0	ETA0 shift modification factor for stress effect (>0)	1.0	

Table 39 New Parameters introduced with BSIM4.3.0 (continued)

Parameter	Description	Default Value	Unit
	Unified Current Saturation		
LAMBDA	Velocity overshoot coefficient If not given or ≤ 0, velocity overshoot will be turned off!	2.0E-5	m/s
VTL	Thermal velocity If not given or ≤ 0, source end thermal velocity limit will be turned off!	2.0E-5	m/s
LC	Velocity back scattering coefficient (~5E-9m at room temperature)	0.0	m
XN	Second velocity back scattering coefficient	3.0	
	Temperature Model		
TEMPMOD	Temperature mode selector TEMPMOD=0: original temperature model TEMPMOD=1: new temperature model	0	
	Holistic Thermal Noise		
RNOIA	Thermal noise coefficient	0.577	
RNOIB	Thermal noise coefficient	0.37	

Main Model Parameters

Table 40 Main Model Parameters

Parameter	Description	Default Value	Unit
	Process related Parameters		
EPSROX	relative gate dielectric constant	3.9 (SiO_2)	-
TOXE	Electrical gate equivalent oxide thickness	3E-9	m
TOXP	Physical gate equivalent oxide thickness	TOXE	-
TOXM	Gate oxide thickness at which parameters are extracted	TOXE	-

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
DTOX	defined as TOXE-TOXP	0.0	m
XJ	Source / Drain junction depth	150E-9	m
GAMMA1	Body-effect coefficient near the surface	$\gamma = \frac{\sqrt{2q \cdot \varepsilon_{si} \cdot NDEP}}{C_{oxe}}$	$V^{1/2}$
GAMMA2	Body-effect coefficient in the substrate	$\gamma = \frac{\sqrt{2q \cdot \varepsilon_{si} \cdot NSUB}}{C_{oxe}}$	$V^{1/2}$
NGATE	Poly Si-gate doping concentration	0.0	cm^{-3}
NDEP	Channel doping concentration at depletion edge for zero body bias	If NDEP is not given but GAMMA1 is given: $NDEP = \frac{(\gamma_1 \cdot C_{oxe})^2}{2 \cdot q \cdot \varepsilon_{Si}}$ If both are not given: NDEP=1E17	cm^{-3}
NSUB	Substrate doping concentration	6E16	cm^{-3}
NSD	Source / Drain doping concentration	1e20	cm^{-3}
XT	Doping depth	1.55E-7	V
VBX	V_{bs} at which depletion region equals XT	$VBX = \Phi_S - \frac{q \cdot NDEP \cdot XT^2}{2 \cdot \varepsilon_{si}}$	V
RSH	Sheet resistance	0.0	$\Omega/(sq)$
RSHG	Gate electrode sheet resistance	0.1	$\Omega/(sq)$

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
	Threshold Voltage		
VFB	Flatband voltage	-1.0	V
VTH0	Long channel threshold voltage at V_{bs} = 0	NMOS: 0.7 PMOS: -0.7	V
PHIN	Non-uniform vertical doping effect on surface potential	0.0	V
K1	First-order body effect coefficient	0.5	$V^{0.5}$
K2	Second-order body effect coefficient	0.0	-
K3	Narrow width coefficient	80.0	-
K3B	Body effect coefficient of K3	0.0	1/V
W0	Narrow width parameter	2.5E-6	m
LPE0	Lateral non-uniform doping parameter at V_{bs} = 0	1.74e-7	m
LPEB	Lateral non-uniform doping effect on K1	0.0	m
VBM	Maximum applied body bias in VTH0 calculation.	-3.0	V
DVT0	First coefficient of short-channel effect on VTH	2.2	-
DVT1	Second coefficient of short-channel effect on VTH	0.53	-
DVT2	Body-bias coefficient of short-channel effect on VTH	-0.032	1/V
DVTP0	First coefficient of drain-induced V_{th} shift for long-channel pocket devices	0.0	m
DVTP1	Second coefficient of drain-induced V_{th} shift for long-channel pocket devices	0.0	V
DVT0W	First coefficient of narrow-width effect on VTH for small channel length	0.0	-
DVT1W	Second coefficient of narrow-width effect on VTH for small channel length	5.3E6	1/m
DVT2W	Body-bias coefficient of narrow-width effect on VTH for small channel length	-0.032	1/V
ETA0	DIBL coefficient in the subthreshold region	0.08	-

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
ETAB	Body-bias for the subthreshold DIBL effect	-0.07	1/V
DSUB	DIBL coefficient exponent in subthreshold region	DROUT	-
	Mobility		
U0	Low-field mobility	NMOS: 670 PMOS: 250	$cm^2/(Vs)$
UA	First-order mobility degradation coefficient due to vertical field	MOBMOD=0 and 1: 1E-9 MOBMOD=2: 1e-15	m/V
UB	Second-order mobility degradation coefficient	1E-19	$(m/V)^2$
UC	Coefficient of the body-bias effect of mobility degradation	MOBMOD=1: -0.0465 MOBMOD=0 and 2: 0.0465E-9	1/V m/V^2
EU	Exponent for mobility degradation of MOBMOD = 2	NMOS: 1.67 PMOS: 1.0	-
	Drain current		
VSAT	Saturation velocity	8.0E4	m/s
A0	Bulk charge effect coefficient	1.0	-
A1	First non-saturation effect factor	0.0	1/V
A2	Second non-saturation effect factor	1.0	-
AGS	Coefficient of V_{gs} dependence of bulk charge effect	0.0	1/V
B0	Bulk charge effect coeff. for channel width	0.0	m
B1	Bulk charge effect width offset	0.0	m
KETA	Body-bias coefficient of the bulk charge effect	-0.047	1/V

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
	Subthreshold region		
VOFF	Offset voltage in subthreshold region for large W and L	-0.08	V
VOFFL	Channel length dependence of VOFF	0.0	mV
MINV	V_{gsteff} fitting parameter for moderate inversion condition	0.0	-
NFACTOR	Subthreshold swing factor	1.0	-
CIT	Interface trap capacitance	0.0	F/m^2
CDSC	Drain-Source to channel coupling capacitance	2.4E-4	F/m^2
CDSCB	Body-bias coefficient of CDSC	0.0	F/Vm^2
CDSCD	Drain-bias coefficient of CDSC	0.0	F/Vm^2
	Drain-Source resistance		
RDSW	Zero bias LDD resistance per unit width for RDSMOD = 0	200	$\Omega\ (\mu m)^{WR}$
RDSWMIN	LDD resistance per unit width at high V_{gs} and zero V_{bs} for RDSMOD = 0	0.0	$\Omega\ (\mu m)^{WR}$
RDW	Zero bias LDD drain resistance per unit width for RDSMOD = 1	100	$\Omega\ (\mu m)^{WR}$
RDWMIN	Zero bias LDD drain resistance per unit width at high V_{gs} and zero V_{bs} for RDSMOD = 1	0.0	$\Omega\ (\mu m)^{WR}$
RSW	Zero bias LDD source resistance per unit width for RDSMOD = 1	100	$\Omega\ (\mu m)^{WR}$
RSWMIN	Zero bias LDD resistance per unit width at high V_{gs} and zero V_{bs} for RDSMOD = 1	0.0	$\Omega\ (\mu m)^{WR}$
WR	Channel width dependence parameter of LDD resistance	1.0	-
PRWB	Body bias coefficient of LDD resistance	0.0	$V^{-0.5}$
PRWG	Gate bias dependence of LDD resistance	1.0	1/V
NRS	Number of source diffusion squares	1.0	-
NRD	Number of drain diffusion squares	1.0	-

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
	Channel geometry		
WINT	Channel width offset parameter	0.0	m
WL	Coeff. of length dependence for width offset	0.0	m^{WLN}
WLN	Power of length dependence for width offset	1.0	-
WW	Coeff. of width dependence for width offset	0.0	m^{WWN}
WWN	Power of width dependence for width offset	1.0	-
WWL	Coeff. of length and width cross term for width offset	0.0	$m^{WLN+WWN}$
LINT	Channel length offset parameter	0.0	m
LL	Coeff. of length dependence for length offset	0.0	m^{LLN}
LLN	Power of length dependence for length offset	1.0	-
LW	Coeff. of width dependence for length offset	0.0	m^{LWN}
LWN	Power of width dependence for length offset	1.0	-
LWL	Coeff. of length and width cross term for length offset	0.0	$m^{LWN+LLN}$
LLC	Coefficient of length dependence for CV channel length offset	LL	-
LWC	Coefficient of width dependence for CV channel length offset	LW	-
LWLC	Coefficient of length and width cross term dependence for CV channel length offset	LWL	-
WLC	Coefficient of length dependence for CV channel width offset	WL	-
WWC	Coefficient of width dependence for CV channel width offset	WW	-
WWLC	Coefficient of length and width cross term dependence for CV channel width offset	WWL	-
LMIN	Minimum channel length	0.0	m
LMAX	Maximum channel length	1.0	m
WMIN	Minimum channel width	0.0	m
WMAX	Maximum channel width	1.0	m
DWG	Coefficient of gate bias dependence of W_{eff}	0.0	m/V

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
DWB	Coefficient of substrate bias dependence of W_{eff}	0.0	$m/V^{0.5}$
	Output resistance		
PCLM	Channel length modulation parameter	1.3	-
PDIBL1	First output resistance DIBL effect parameter	0.39	-
PDIBL2	Second output resistance DIBL effect parameter	8.6m	-
PDIBLB	Body bias coefficient of output resistance DIBL effect	0.0	1/V
DROUT	Channel-length dependence coefficient of the DIBL effect on output resistance	0.56	-
PSCBE1	First substrate current induced body-effect parameter	4.24E8	V/m
PSCBE2	Second substrate current induced body-effect coefficient	1.0E-5	m/V
PVAG	Gate-bias dependence of Early voltage	0.0	-
FPROUT	Effect of pocket implant on R_{out} degradation	0.0	$V/m^{0.5}$
PDITS	Impact of drain-induced V_{th} shift on R_{out}	0.0	V^{-1}
PDITSL	Channel-length dependence of drain-induced V_{th} shift on R_{out}	0.0	m^{-1}
PDITSD	V_{ds} dependence of drain-induced V_{th} shift on R_{out}	0.0	V^{-1}
ALPHA0	First impact ionization parameter	0.0	Am/V
ALPHA1	Length dependent substrate current parameter	0.0	A/V
BETA0	Second impact ionization parameter	30	V
	Gate-Induced Drain Leakage model		
AGIDL	Pre-exponential coefficient for GIDL	0.0	mho (1/Ohm)
BGIDL	Exponential coefficient for GIDL	2.3e9	V/m
CGIDL	Parameter for body-bias effect on GIDL	0.5	V^3
EGIDL	Fitting parameter for band bending for GIDL	0.8	V

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
	Gate Dielectric Tunneling Current		
AIGBACC	Parameter for I_{gb} in accumulation	0.43	$\frac{\sqrt{(Fs^2)/g}}{m}$
BIGBACC	Parameter for I_{gb} in accumulation	0.054	$\frac{\sqrt{(Fs^2)/g}}{m \cdot V}$
CIGBACC	Parameter for I_{gb} in accumulation	0.075	1/V
NIGBACC	Parameter for I_{gb} in accumulation	1.0	-
AIGBINV	Parameter for I_{gb} in inversion	0.35	$\frac{\sqrt{(Fs^2)/g}}{m}$
BIGBINV	Parameter for I_{gb} in inversion	0.03	$\frac{\sqrt{(Fs^2)/g}}{m \cdot V}$
CIGBINV	Parameter for I_{gb} in inversion	0.006	1/V
EIGBINV	Parameter for I_{gb} in inversion	1.1	V
NIGBINV	Parameter for I_{gb} in inversion	3.0	-
AIGC	Parameter for I_{gcs} and I_{gcd}	NMOS: 0.054 PMOS: 0.31	$\frac{\sqrt{(Fs^2)/g}}{m}$
BIGC	Parameter for I_{gcs} and I_{gcd}	NMOS: 0.054 PMOS: 0.024	$\frac{\sqrt{(Fs^2)/g}}{m \cdot V}$
CIGC	Parameter for I_{gcs} and I_{gcd}	NMOS: 0.075 PMOS: 0.03	V

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
AIGSD	Parameter for I_{gs} and I_{gd}	NMOS: 0.43 PMOS: 0.31	$\frac{\sqrt{(Fs^2)/g}}{m}$
BIGSD	Parameter for I_{gs} and I_{gd}	NMOS: 0.054 PMOS: 0.024	$\frac{\sqrt{(Fs^2)/g}}{m \cdot V}$
CIGSD	Parameter for I_{gs} and I_{gd}	NMOS: 0.075 PMOS: 0.03	V
DLCIG	Source/Drain overlap length for I_{gs} and I_{gd}	LINT	-
NIGC	Parameter for I_{gcs}, I_{gcd}, I_{gs} and I_{gd}	1.0	-
POXEDGE	Factor for gate oxide thickness in source/drain overlap regions	1.0	-
PIGCD	Vgs dependence of I_{gcs} and I_{gcd}	1.0	-
NTOX	Exponent for the gate oxide ratio	1.0	-
TOXREF	Nominal gate oxide thickness for gate direct tunneling model	3E-9	m
	Diode Characteristics		
IJTHSREV	(Source) Limiting current in reverse bias region	IJTHSREV =0.1	A
IJTHDREV	(Drain)	IJTHDREV =IJTHSREV	
IJTHSFWD	(Source) Limiting current in forward bias region	IJTHSFWD =0.1	A
IJTHDFWD	(Drain)	IJTHDFWD =IJTHSFWD	
XJBVS	(Source) Fitting parameter for diode breakdown	XJBVS=1.0	-
XJBVD	(Drain)	XJBVD =XJBVS	
BVS	(Source) Breakdown voltage	BVS=10.0	V
BVD	(Drain)	BVD=BVS	

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
JSS JSD	(Source) Bottom junction reverse saturation current density (Drain)	JSS=1.0e-4 JSD=JSS	A/m²
JSWS JSWD	Isolation-edge sidewall reverse saturation current density	JSWS =0.0 JSWD =JSWS	A/m
JSWGS JSWGD	Gate-edge sidewall reverse saturation current density	JSWGS=0.0 JSWGD=JSWGS	A/m
CJS CJD	Bottom junction capacitance per unit area at zero bias	CJS=5.0e-4 CJD=CJS	F/m²
MJS MJD	Bottom junction capacitance grating coefficient	MJS=0.5 MJD=MJS	-
MJSWS MJSWD	Isolation-edge sidewall junction capacitance grading coefficient	MJSWS =0.33 MJSWD =MJSWS	-
CJSWS CJSWD	Isolation-edge sidewall junction capacitance per unit area	CJSWS= 5.0e-10 CJSWD=CJSWS	F/m
CJSWGS CJSWGD	Gate-edge sidewall junction capacitance per unit length	CJSWGS =CJSWS CJSWGD =CJSWS	-
MJSWGS MJSWGD	Gate-edge sidewall junction capacitance grading coefficient	MJSWGS=MJSWS MJSWGD=MJSWS	-
PBS	Source bottom junction built-in potential	PBS=1.0	V
PBD	Drain bottom junction built-in potential	PBD=PBS	V
PBSWS	Isolation-edge sidewall junction built-in potential of source junction	PBSWS =1.0	V
PBSWD	Isolation-edge sidewall junction built-in potential of drain junction	PBSWD=PBSWS	V

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
PBSWGS	Gate-edge sidewall junction built-in potential of source junction	PBSWGS =PBSWS	V
PBSWGD	Gate-edge sidewall junction built-in potential of drain junction	PBSWGD=PBSWS	V
	Capacitance		
XPART	Charge partitioning parameter	0.0	-
CGSO	Non LDD region gate-source overlap capacitance per unit W	calculated, see Overlap Capacitance Model	F/m
CGDO	Non LDD region gate-drain overlap capacitance per unit W	calculated, see Overlap Capacitance Model	F/m
CGBO	Gate-bulk overlap capacitance per unit L	0.0	F/m
CGSL	Light doped gate-source region overlap capacitance	0.0	F/m
CGDL	Light doped gate-drain region overlap capacitance	0.0	F/m
CKAPPAS	Coefficient of bias-dependent overlap capacitance on source side	0.6	V
CKAPPAD	Coefficient of bias-dependent overlap capacitance on drain side	CKAPPAS	V
CF	Fringing field capacitance	$\frac{2 \cdot EPSROX \cdot \varepsilon_0}{\pi} \cdot \log\left(1 + \frac{4e-7}{TOXE}\right)$	F/m
CLC	Constant term for the short channel model	0.1E-7	m
CLE	Exponential term for the short channel model	0.6	-
DLC	Length offset fitting parameter for CV model	LINT	m
DWC	Width offset fitting parameter for CV model	WINT	m
VFBCV	Flatband voltage parameter for CAPMOD = 0	-1.0	V
NOFF	CV parameter in $V_{gsteff,CV}$ for weak to strong inversion	1.0	-
VOFFCV	CV parameter in $V_{gsteff,CV}$ for weak to strong inversion	0.0	V
ACDE	Exponential coefficient for charge thickness in accumulation and depletion regions in CAPMOD=2	1.0	m/V

Table 40 Main Model Parameters (continued)

Parameter	Description	Default Value	Unit
MOIN	Coefficient for the gate-bias dependent surface potential	15.0	-

Temperature Modeling Parameters

Table 41 Temperature Modeling Parameters

Parameter	Description	Default Value	Unit
TNOM	Parameter extraction temperature	27	°C
UTE	Mobility temperature coefficient	-1.5	-
KT1	Threshold voltage temperature coefficient	-0.11	V
KT1L	Channel length dependence of KT1	0.0	Vm
KT2	Threshold voltage temperature coefficient	0.022	-
UA1	Temperature coefficient for UA	1E-9	m/V
UB1	Temperature coefficient for UB	-1E-18	$(m/V)^2$
UC1	Temperature coefficient for UC	MOBMOD=1: 0.056 MOBMOD=0 and 2: 0.056E-9	1/V m/V^2
PRT	Temperature coefficient for RDSW	0.0	Ω m
AT	Saturation velocity temperature coefficient	3.3E4	m/s
NJS	Emission coefficient for Source junction	1.0	-
NJD	Emission coefficient for Drain junction	NJS	-
XTIS	Junction current temperature exponent coefficient of source body junction	3.0	-
XTID	Junction current temperature exponent coefficient of drain body junction	XTIS	-
TPB	Temperature coefficient for PB	0.0	V/K
TPBSW	Temperature coefficient for PBSW	0.0	V/K
TPBSWG	Temperature coefficient for PBSWG	0.0	V/K

Table 41 Temperature Modeling Parameters (continued)

Parameter	Description	Default Value	Unit
TCJ	Temperature coefficient for CJ	0.0	1/K
TCJSW	Temperature coefficient for CJSW	0.0	1/K
TCJSWG	Temperature coefficient for CJSWG	0.0	1/K

Flicker Noise Model Parameters

Table 42 Flicker Noise Model Parameters

Parameter	Description	Default Value	Unit
NOIA	Flicker noise parameter A	NMOS: 6.25e41 PMOS: 6.188e40	$(eV)^{-1}\,s^{1-EF}m^{-3}$
NOIB	Flicker noise parameter B	NMOS: 3.125e26 PMOS: 1.5e25	$(eV)^{-1}\,s^{1-EF}m^{-1}$
NOIC	Flicker noise parameter C	8.75	$(eV)^{-1}\,s^{1-EF}m$
EM	Saturation field	4.1e7	V/m
AF	Flicker noise exponent	1.0	-
EF	Flicker noise frequency exponent	1.0	-
KF	Flicker noise coefficient	0.0	$A^{2-EF}\,s^{1-EF}F$
NTNOI	Noise factor for short-channel devices for TNOIMOD=0 only	1.0	-
TNOIA	Coefficient of channel-length dependence of total channel thermal noise	1.5	-
TNOIB	Channel-length dependence parameter for channel thermal noise partitioning	3.5	-

High-Speed / RF Model Parameters

Table 43 High-Speed/RF Model Parameters

Parameter	Description	Default Value	Unit
XRCRG1	Parameter for distributed channel resistance effect for both intrinsic input resistance and charge-deficit NQS models	12.0	-
XRCRG2	Parameter to account for the excess channel diffusion resistance for both intrinsic input resistance and charge-deficit NQS models	1.0	-
RBPB	Resistance connected between bNodePrime and bNode	50.0	Ohm
RBPD	Resistance connected between bNodePrime and dbNode	50.0	Ohm
RBPS	Resistance connected between bNodePrime and sbNode	50.0	Ohm
RBDB	Resistance connected between dbNode and bNode	50.0	Ohm
RBSB	Resistance connected between sbNode and bNode	50.0	Ohm
GBMIN	Conductance in parallel with each of the five substrate resistances to avoid potential numerical instability due to unreasonably too large a substrate resistance	1.0e-12	mho

Layout-Dependent Parasitics Model Parameters

Table 44 Layout-Dependent Parasitics Model Parameters

Parameter	Description	Default Value	Unit
DMCG	Distance from S/D contact center to the gate edge	0.0	m
DMCI	Distance from S/D contact center to the isolation edge in the channel length direction	DMCG	-
DMDG	Same as DMCG but for merged device only	0.0	m
DMCGT	DMCG of test structures	0.0	m
NF	Number of device fingers	1.0	-
DWJ	Offset of the S/D junction width (in CV model)	DWC	-
MIN	Whether to minimize the number of drain or source diffusions for even number fingered devices	0.0	-
XGW	Distance from the gate contact to the channel edge	0.0	m
XGL	Offset of the gate length due to variations in patterning	0.0	m
XL	Channel length offset due to mask/etch effect	0.0	m
XW	Channel width offset due to mask/etch effect	0.0	m
NGCON	Number of gate contacts	1.0	-

Model Selection Flags

Table 45 Model Selection Flags

Parameter	Values	Type of Model
LEVEL	14	BSIM4 model selector (in UCB SPICE3)
VERSION	4.3.0	Model version number
BINUNIT	0, ***1***	Binning unit selector
PARAMCHK	0, ***1***	Switch for Parameter value check (*Parameters checked*)
MOBMOD	***0***, 1, 2	Mobility model (*same as in BSIM3v3.2*)
RDSMOD	***0***, 1	Bias-dependent source/drain resistance model selector (*internal Rds(V)*)
IGCMOD	***0***, 1	Gate-to-channel tunneling current model selector (I_{gc}, I_{gs}, I_{gd} *are off*)
IGBMOD	***0***, 1	Gate-to-substrate tunneling current model selector (I_{gb} *is off*)
CAPMOD	0, 1, ***2***	Capacitance model selector (*single-equation and charge-thickness model*)
RGATEMOD	***0***, 1, 2, 3	Gate resistance model selector (*no gate resistance*)
RBODYMOD	***0***, 1	Substrate resistance network model selector (*network off*)
TRNQSMOD	***0***, 1	Charge-deficit transient non quasi static model selector (*charge-deficit model off*)
ACNQSMOD	***0***, 1	Charge-deficit AC small signal non quasi static model selector (*charge-deficit model off*)
FNOIMOD	0, ***1***	Flicker noise model selector (*unified physical flicker noise model is used*)
TNOIMOD	*0*, 1	Thermal noise model selector (*charge-based thermal noise model*)
DIOMOD	0, ***1***, 2	Asymmetric source/drain junction diode IV model selector (*Junction diodes are modeled breakdown-free*)
PERMOD	0, ***1***	PS / PD parameters include gate-edge perimeter (*including the gate-edge perimeter*)
GEOMOD	***0***, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Geometry-dependent parasitics model selector - specify how the end S/D-diffusions are connected (*isolated*)

Table 45 Model Selection Flags

Parameter	Values	Type of Model
RGEOMOD	***0***, 1, 2, 3, 4, 5, 6, 7, 8	S/D diffusion resistance and contact model selector: specifying the end S/D contact type (point, wide or merged) and how S/D parasitic resistance is computed (*no S/D diffusion resistance*)

NOTE Underlined values in bold italics are defaults, underlined comments in italics (in brackets) are valid for default model selector values.

References

1 BSIM4.3.0 Manual, University of California at Berkeley, Copyright © 2003 The Regents of the University of California. See the web site of the device research group at UCB. You can download the manual from the Internet, using the following Web address: http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html

2 "Characterization System for Submicron CMOS Technologies," JESSI Reports AC41 94-1 through 94-6

3 C. Enz, "MOS Transistor Modeling for RF IC Design", Silicon RF IC: Modeling and Simulation Workshop, Lausanne 2000

4 T. Gneiting, "BSIM4, BSIM3v3 and BSIMSOI RF MOS Modeling", RF Modeling and Measurement Workshop, European Microwave Week, Paris 2000

5 William Liu, "Mosfet Models for Spice Simulation: Including BSIM3v3 and BSIM4", John Wiley & Sons, January 2001

6 M. J. Deen (Ed.), T.A.Fjeldly, "CMOS RF Modeling, Characterization and Applications", Worldscientific, Co-authors: F.Sischka and T.Gneiting

Acknowledgements

The BSIM4 model was developed by the UC Berkeley BSIM Device Research Group of the Department of Electrical Engineering and Computer Science, University of California, Berkeley and copyrighted by the University of California.

6 UCB GaAs MESFET Characterization

UCB GaAs MESFET Model 429
Simulators 430
Model Parameters 433
Test Instruments 435
Measuring and Extracting 436
Extraction Algorithms 446
References 447

This chapter describes the UCB GaAs MESFET transistor model supported by SPICE. Descriptions of model setup, instrument connections, and model parameters are included as well as test instrument information. Procedures are included for extracting AC and DC model parameters from GaAs MESFET transistors using the UCB GaAs MESFET Model. These model parameters describe the operating characteristics of the device under test (DUT) and can be derived from either simulated or direct measurements of the DUT.

The IC-CAP UCB GaAs MESFET modeling module provides setups that can be used for general measurement and model extraction for GaAs technology. The IC-CAP system offers the flexibility to modify any measurement or simulation specification.

The model extractions provided are also intended for general GaAs IC processes. If you have another method of extracting specific model parameters, you can do so with the Program function or by writing a function in C and linking it to the

Function List. Details on the Program transform and writing user-defined C language routines are explained in Chapter 9, "Using Transforms and Functions," in the *IC-CAP User's Guide*.

The model presented here has been enhanced with the inclusion of the series inductors and the gate resistor. Both of these are implemented as circuit elements of the overall subcircuit. This is an example of one method you might use to customize your own model.

UCB GaAs MESFET Model

The UCB GaAs MESFET model is contained in an IC-CAP example file *UCBGaas.mdl*. The file consists of a single level model that is based on a model developed at Raytheon and implemented in UCB's SPICE3 simulator [1][2]. In this model, drain current is proportional to the square of the gate-to-source voltage multiplied by the expansion series of the hyperbolic tangent of the drain-to-source voltage.

The model defines total gate junction capacitance and takes into account the FET symmetry and carrier velocity saturation. Unlike the SPICE3 version, the IC-CAP model includes gate, source, and drain inductances, and gate resistance. These components are extracted in IC-CAP as external resistance and inductances to improve the accuracy of the model.

Simulators

This model is supported by SPICE3. HPSPICE provides only DC analysis capability for this model.

NOTE Simulators are provided as a courtesy to IC-CAP users; they are not supported by Agilent Technologies.

The SPICE3 default nominal temperature is 27°C. Set the *TNOM* variable under the *Utilities* menu to force another temperature.

SPICE3 Simulators

The general form for the SPICE3 statement that calls the UCB GaAs MESFET model is

ZXXXXXXX ND NG NS MNAME

where

ZXXXXXXX	indicates MESFET device name (any name that begins with Z)
ND	indicates drain node number
NG	indicates gate node number
NS	indicates source node number
MNAME	indicates model name

An example of this call is

```
Z1 7 2 3 ZM1 OFF
```

The SPICE3 syntax of the *.MODEL* definition is

.MODEL MNAME TYPE PNAME1=PVAL1 PNAME2=PVAL2 ...

where

MNAME	indicates model name

TYPE	indicates NMF (N-channel MESFET) or PMF (P-channel MESFET)
PNAME#	indicates UCB GaAs MESFET parameter name
PVAL#	indicates parameter value of *PNAME#*

HPSPICE Simulators

The general form for the HPSPICE statement that calls the UCB GaAs MESFET model is

JXXXXXXX ND NG NS MNAME

where

JXXXXXXX	indicates MESFET device name (any name that begins with J)
ND	indicates drain node number
NG	indicates gate node number
NS	indicates source node number
MNAME	indicates model name

An example of this statement is

```
J1 5 1 2 MODJ
```

The HPSPICE syntax for the .MODEL definition is

.MODEL MNAME RCAY TYPE PNAME1=PVAL1 PNAME2=PVAL2 ...

where

MNAME	indicates model name
RCAY	key word specifying a GaAs MESFET model
TYPE	indicates NJF (N-channel MESFET) or PJF (P-channel MESFET)
PNAME#	indicates UCB GaAs MESFET parameter name
PVAL#	indicates parameter value of *PNAME#*

The parameter *MODEL* in the .MODEL description must be set.

MODEL = 3 indicates UCB GaAs MESFET model

IC-CAP allows you to assign node names to node numbers, simplifying references to nodes (by a meaningful name). For more information, refer to "Assigning Node Names" in the *Reference*.

Model Parameters

UCB GaAs MESFET model parameters are described in the following table. Setup attributes are listed in Table 47.

Table 46 UCB GaAs MESFET Model Parameter

Name	Description	Default
Inductance and Resistance Parameters		
LD	Drain inductance. Specifies external drain inductance.	0 Henry
LG	Gate inductance. Specifies external gate inductance.	0 Henry
LS	Source inductance. Specifies external source inductance.	0 Henry
RD	Drain resistance. Specifies external drain resistance.	0 Ohm
Diode Parameters		
IS	Diode reverse saturation current. Models gate-drain and gate-source current.	1×10^{-14} Amp
PB	Gate junction potential. Models built-in potentials of gate-source and gate-drain regions.	1V
XN	Diode emission coefficient. Models emission coefficient of an ideal diode.	1
DC Parameters		
ALPHA	Saturation voltage parameter. Specifies voltage at which drain current reaches saturation. The Vds coefficient in the tanh function.	$2.0V^{-1}$
B	Doping profile parameter. Models intrusion of doping profile into the insulating substrate.	$0.3V^{-1}$
BETA	Transconductance parameter. Defines transconductance in the saturation or linear operating regions.	1×10^{-4} $A\cdot V^{-2}$

Table 46 UCB GaAs MESFET Model Parameter

Name	Description	Default
LAMBDA	Channel length modulation parameter. Models finite output conductance of a MESFET in the saturation region.	$0V^{-1}$
VTO	Zero bias threshold voltage. Models gate turn-on voltage.	0V
Capacitance Parameters		
CGD	Zero bias gate-drain capacitance	0 Farad
CGS	Zero bias gate-source capacitance	0 Farad

Table 47 UCB GaAs Model Setup Attributes

DUT / Setup	Inputs	Out-puts	Transform	Function	Extractions
ac / s_at_f	vg, vd, vs, freq	s	extract_L_ and_R	GAASAC_I_and_r	LD, LG, LS, RD, RG, RS
ac / s_vs_f	vg, vd, vs, freq	s	extract_CV	GAASCV_cgs_cgd	CGD,CGS
dc / igvg_0v[sd]	vg, v[sd]	ig	extract	GAASDC_lev1	PB, IS, XN
			optim1		PB, IS, XN
dc / idvg_hi_vd	vg, vd, vs	id, ig	none		
dc / idvd_vg	vd, vg, vs	id, ig	extract	GAASDC_lev2	VTO, BETA, ALPHA, LAMBDA, B
			optim1	Optimize	VTO, BETA, ALPHA, LAMBDA
			optim2	Optimize	VTO, BETA, B

Test Instruments

The HP 4141, Agilent 4142, HP 4145, Agilent 4155, or Agilent 4156 can be used to derive DC model parameters from measured DC voltage and current characteristics.

The Agilent 8510, Agilent 8753, or HP 8702 (with an HP 41xx instrument) can be used to derive capacitance and inductance model parameters from S-parameter measurements.

Instrument-to-Device Connections

When the device is installed in a test fixture, verify the correct connection of device nodes by checking the specifications in the setup tables. Table 48 is a cross-reference of the connections between the terminals of a typical MESFET device and various measurement units. These connections and measurement units are defined in the *UCBGaas.mdl* example file.

Input and output tables in the various setups use abbreviations D (drain), G (gate), and S (source) for the MESFET device nodes. These nodes are defined in the Circuit folder.

Measurement units (abbreviated as follows) are defined in Hardware Setup.

SMU# for DC measurement units
NWA for network analyzer units

Table 48 Instrument-to-Device Connections

DUT	Source	Gate	Drain	Comments
dc	SMU3	SMU2	SMU1	
ac	Ground	SMU2 NWA (port 1)	SMU1 NWA (port 2)	Calibrate for reference plane

Notes:
1. DUT is the name of the DUT as specified in DUT-Setup.
2. Example: DUT *dc* has the DC measurement unit SMU1 connected to its drain, SMU2 connected to its gate, and SMU3 connected to its source.

Measuring and Extracting

This section provides general information as well as procedures for performing measurements and extractions of MESFET devices.

Measurement and Extraction Guidelines

The following guidelines are provided to help you achieve more successful model measurements and extractions.

Setting Instrument Options

Before starting a measurement, you can quickly verify instrument option settings. Save the current instrument option settings by saving the example file to *<file_name>.mdl* from the model window. Some of the Instrument Options specify instrument calibration. For the most accurate results, calibrate the instruments before taking IC-CAP measurements.

- DC measurements are generally taken with *Integration Time* = *Medium*.
- CV measurements in the femtofarad region usually require *High Resolution* = *Yes* and *Measurement Freq (kHz)* = *1000*.
- When taking AC measurements with a network analyzer, several instrument settings are critical. In addition, the calibration must be performed on structures that have similar impedances as the stray parasitics of the DUT.
- Input power to the device is typically –10 to –30dBm (after port attenuation).
- Setting the averaging factor to the 2-to-4 range reduces measurement noise.
- Because IC-CAP requires the instrument to perform error correction, set *Use Internal Calibration* to *Yes*.

Experiment with the other network analyzer options to obtain the best results with specific devices.

Measuring Instruments

Ensure that the measuring instruments (specified by unit names in the input and output tables) are correctly connected to the DUT. Refer to Table 48 for a list of nodes and corresponding measurement units. The quality of the measuring equipment (instruments, cables, test fixture, transistor sockets, and probes) can influence the noise level in the measurements.

Ensure that all characteristics of the measurement stimulus and corresponding measured response are specified in the respective input and output tables.

Calibration

For some measurements the instruments or test hardware must be calibrated to remove non-device parasitics from the DUT. For MESFET devices, stray capacitance due to probe systems, bond pads, and so on should be calibrated out prior to each measurement.

For high-frequency 2-port measurements with a network analyzer, the reference plane of the instrument must be calibrated out to the DUT. IC-CAP relies on the internal calibration of the instruments for full error-corrected data. It is critical that calibration using *OPEN, SHORT, THRU,* and *50 ohm* loads be properly done.

Extracting Model Parameters

For a given setup, you can find the extraction transforms in the Extract/Optimize folder. IC-CAP's extraction algorithms exist as functions; choose **Browse** to list the functions available for a setup.

When the *extract* command is selected from the setup, all extractions in the setup are performed in the order listed in the setup. This order is usually critical to proper extraction performance. Extractions are typically completed instantly and the newly extracted model parameter values are placed in Model Parameters.

IC-CAP provides setups for two extraction methods. In general, you only need to perform one of the methods in order to extract parameters.

Simulating Device Response

Simulation uses model parameter values currently in the Parameters table. A SPICE deck is created and the simulation performed. The output of the SPICE simulation is then read into IC-CAP as simulated data.

SPICE3 is the only simulator fully compatible with the IC-CAP *UCBGaas.mdl* configuration file is. You can also use the HPSPICE simulator if you modify the parameter names to match it. DC simulations generally run much faster than cv and AC simulations.

If simulated results are not as expected, use the simulation debugger (in the *Tools* menu) to examine the input and output simulation files. The output of manual simulations is not available for further processing by IC-CAP functions such as transforms and plots.

Displaying Plots

The Display Plot function displays all graphical plots defined in a setup. The currently active graphs are listed under the Plots folder in each setup. View the plots for agreement between measured and simulated data. Measured data is displayed as a solid line; simulated data is displayed as a dashed or dotted line.

Optimizing Model Parameters

Optimization of model parameters improves the agreement between measured and simulated data. An optimize transform whose *Extract Flag* is set to *Yes* is automatically called after any extraction that precedes it in the transform list.

Optimizing AC parameters can be very time consuming because of the number of SPICE simulations required.

Extraction Procedure Overview

This section describes the general procedure for extracting model parameter data from the UCB GaAs MESFET. The general procedure applies to all types of parameters. Differences between extracting one type and another are primarily in the types of instruments, test setups, and transforms used.

Parameters are extracted from measured or simulated data. Measured data is data taken directly from instruments connected to the DUT inputs and outputs. Simulated data are results from the simulator. Once measured and simulated data have been obtained, each data set can be plotted and compared in the *Plot* window.

The general extraction procedure is summarized next, starting with the measurement process.

1. Install the device to test in a test fixture and connect the test instruments.
2. Ensure the test fixture, signal source and measuring instruments, and workstation are physically and logically configured for the IC-CAP system.
3. Choose **File > Open > Examples**. Select **UCBGaas.mdl** and choose **OK**. Select **Open** to load the file and open the model window. Choose **OK**.

   When the UCBGaas model window opens you are ready to begin measurement and extraction operations.
4. Enter the variable name `EXTR_PAR` at the Model level and enter `NMF1` as its value. This allows the extractions to find the model parameters for the model name *NMF1* within the subcircuit of the model file. This concept is covered in more detail in Chapter 9, “Circuit Modeling.”
5. Select the DUT and setup.
6. Issue the *Measure* command.
7. Issue the *Extract* command.
8. Issue the *Simulate* command.
9. Display the results.

10 Fine tune the extracted parameters if needed by optimizing.

Parameter Measurement and Extraction

The recommended method for extracting UCB GaAs model parameters is presented next. In this extraction, external resistances are extracted from AC data.

NOTE If AC data is not available, an alternative method (described in "Alternate Extraction Method" on page 442) uses the Fukui technique [3] for extracting the resistances from DC data. Use the alternative method only if AC data is not available; the recommended method produces parameters that are more precise.

NOTE The UCB GaAs MESFET model extractions and Special Functions in IC-CAP only support the model as defined by the UCB SPICE3 implementation. There is a difference in several model parameter names from the Curtice model and the UCB model as they are implemented in the HPSPICE simulator. Valid model parameter names are listed in Table 46.

Parameter extractions are dependent on each other; to ensure accuracy extractions must be done in this order:

Inductances and resistances (AC) External inductance and resistance parameters are extracted from an S-parameter measurement at a single bias setting. The gate of the device is strongly forward-biased to make the device look like a short circuit. The setup *s_at_f* is used to take the measurements and extract the parameters LD, LG, LS, RD, RG, and RS.

Diode parameters (DC) Diode parameters PB, IS, and XN are extracted from data produced by the measurement of Ig versus Vg measured at zero drain voltage, with the source floating. The setup *igvg_0vs* or *igvg_0vd* is used to make the measurements and extractions, depending on whether the gate-source or gate-drain junction is preferred.

Other DC parameters (DC) The remaining DC parameters are extracted using two setups: *idvd_vg* and *idvg_hi_vd*. Use *idvd_vg* to measure Id versus Vg at different gate voltages, then use *idvg_hi_vd* to measure Id versus Vg at a constant drain voltage. Parameters VTO, BETA, ALPHA, LAMBDA, and B are then extracted from the resulting data.

AC (capacitance parameters) The capacitance parameters CGD and CGS are extracted from an S-parameter measurement using the setup *s_vs_f*. The measured data is first corrected using the inductances and resistances extracted in the initial step, then the capacitances are extracted from the corrected data.

Use a network analyzer to make the next set of measurements. S-parameter measurements are highly sensitive—the instrument must be properly calibrated.

1 Place the device to be measured in the test fixture.

NOTE

For the *ac/s_at_f* and *s_vs_f* measurements, the SMUs connected to the network analyzer's port bias connections must correspond to the SMUs in Table 48.

2 Select the **ac/s_at_f** DUT/setup and choose **Measure**.

3 In the ac/s_at_f DUT/setup choose **Extract** to extract inductance and resistance parameters.

4 Select the **ac/s_vs_f** DUT/setup and choose **Measure**. Do not extract the parameters for this setup yet.

5 Disconnect the device from the network analyzer and directly connect it to the appropriate units for DC measurements.

6 Select the **dc/ igvg_0vs** or **igvg_0vd** DUT/setup and choose **Measure**.

7 Repeat Step 6, but choose **Extract** to extract diode parameters.

8 Repeat Step 6, but choose **Optimize.**

9 Select the **dc/idvg_hi_vd** DUT/setup and choose **Measure** to measure Id versus Vg at a constant Vd.

10 Select the **dc/idvd_vg** DUT/setup and perform the measure and extract steps to measure and extract the other DC parameters.

11 Repeat Step 10, but choose **Optimize.**

12 Select the **ac/s_vs_f** DUT/setup and choose **Extract** to extract the capacitances from the data that was measured in step 4.

All model parameters are extracted and their values added to the Parameters table; they can be viewed in the Model Parameters folder.

Alternate Extraction Method

If AC data is not available, IC-CAP supports an alternate method for extracting UCB MESFET model parameters. This procedure uses the *Fukui* technique [3]; external resistances are extracted along with the diode parameters from DC data—this differs from the recommended method. Use this method only if the AC data is not available—this alternate method produces parameters that are less precise than those of the recommended method.

Parameter extractions are dependent on each other; to ensure accuracy extractions must be done in the following order.

Resistance and diode parameters (DC) Using DC measurements only, this procedure uses the *Fukui* algorithm to extract the resistance parameters RD, RG, and RS from DC data (refer to Table 46). Diode parameters PB, IS, and XN are also extracted. The extraction requires the setups listed in the following table.

Table 49 Resistance and Diode Measurement and Extraction Setups for the Alternative Method

idvg_low_vd	Id versus Vg	Small Vd
igvg_0vs	Ig versus Vg	Vs=0, with Drain floating
igvg_0vd	Ig versus Vg	Vd=0, with Source floating

The extraction is performed from the setup *igvg_0vd*. To use the Fukui algorithm, the following inputs must be added to the function *GAASDC_lev1* (*extract* transform).

VG (low Vds) idvg_low_vd/vg
VD (low Vds) idvg_low_vd/vd
ID (low Vds) idvg_low_vd/id.m
VG (D Flt) igvg_0vs/vg
IG (D Flt) igvg_0vd/ig.m

Other DC parameters (DC) Use the recommended method described previously.

Inductance parameters (AC) Use the recommended method described previously. Parameters LD, LG, and LS are extracted from the S-parameter measurement. The same transform also extracts the resistance parameters, overwriting the existing ones as it does so.

Capacitance parameters (AC) Use the recommended method described previously.

The alternate extraction procedure follows.

1 Connect the NWA to extract inductances and capacitances.

2 Place the device to be measured in the test fixture.

NOTE

For the *ac/s_at_f* and *s_vs_f* measurements, the SMUs connected to the network analyzer's port bias connections must correspond to the same SMUs in Table 48.

3 Select the **ac/s_at_f** DUT/setup and choose **Measure**.

4 In the ac/s_at_f DUT/setup, choose **Extract** to extract inductance and resistance parameters.

5 Select the **ac/s_vs_f** DUT/setup and choose **Measure**.

6 In the ac/s_vs_f DUT/setup, choose **Extract** to extract the capacitances from the data that was measured in step 3.

7 Select **dc/igvg_0vs** DUT/setup and choose **Measure**.

8 Repeat Step 7 for dc/idvg_low_vd and dc/igvg_0vd.

9 In the dc/igvg_0vd DUT/setup, choose **Extract** to extract the resistance and diode parameters from the measured data for the three DC setups.

10 Repeat Step 9 but choose **Optimize**.

11 Select the **dc/idvg_hi_vd** DUT/setup and choose **Measure** to measure Id versus Vg at a constant Vd.

12 In the dc/idvd_vg DUT/setup, repeat the Measure and Extract steps to measure and extract the other DC parameters.

13 Repeat step 13 but choose **Optimize**.

Simulating

To simulate any individual setup, choose Simulate with that setup active. Simulations can be performed in any order once all of the model parameters have been extracted.

For more information on simulation, refer to Chapter 6, "Simulating," in the *IC-CAP User's Guide*.

Displaying Plots

To display plots issue the *Display Plot* command from a DUT to display the plots for all setups in that DUT. The Plots use the most recent set of measured and simulated data. Viewing plots is an ideal way to compare measured and simulated data to determine if further optimization would be useful. For more information on Plots, refer to Chapter 10, "Printing and Plotting," in the *IC-CAP User's Guide*.

Optimizing

The optimization operation uses a numerical approach to minimize errors between measured and simulated data. As with the other IC-CAP commands, optimization can be performed at either the DUT or setup level. Optimization is more commonly performed from setups—optimization for all setups under a DUT is rarely required.

Optimization is typically interactive in nature, with the best results obtained when you specify the characteristics of the desired results.

For more information on optimization, refer to Chapter 7, "Optimizing," in the *IC-CAP User's Guide*.

Extraction Algorithms

This section describes the extraction algorithms used for inductance, capacitance, DC, and series resistance parameters of the UCB GaAs MESFET.

Inductance and Resistance Extraction

Inductors and resistors that are in series with each terminal of the MESFET are measured with a network analyzer at a high frequency and with the gate strongly forward-biased. Under these conditions the AC model for each terminal is reduced to a series R-L circuit. Inductors LD, LG, and LS and resistors RD, RG, and RS are extracted simultaneously from the measured impedance at the gate and drain ports.

DC Parameter Extractions

DC extractions are separated between the diode and forward active controlling parameters. Diode parameters PB, IS, and XN are extracted using a method similar to the method used for a silicon diode. The diode is swept over a forward bias and linear least-squares curve fits will produce the built-in potential and forward conduction properties.

The forward active region is extracted from measurements of the MESFET in both the linear and saturated modes of operation. The difference between the equations defining these two regions is a tanh(ALPHA · Vds) multiplier in the linear mode. The threshold can be obtained from a least-squares fit to the linear region. These equations are solved to produce ALPHA, B, BETA, LAMDA, and VTO.

Capacitance Parameter Extractions

The capacitance of the gate-to-drain (CGD) and gate-to-source (CGS) is measured with S-parameters over typical operating frequencies. Because the inductances and series resistance are already known, these capacitances can be extracted from the corrected impedances measured at the gate and drain ports.

References

1 Hermann Statz, Paul Newman, Irl W. Smith, Robert A. Pucel, & Hermann A. Haus. *GaAs FET Device and Circuit Simulation in SPICE,* IEEE Transactions on Electron Devices, Vol. ED-34, No. 2, February 1987.

2 Tom Quarles et al. *SPICE Version 3A7 User's Guide,* University of California, Berkeley.

3 H. Fukui, *Determination of the Basic Device Parameters of a GaAs MESFET*, The Bell System Technical Journal, Vol. 58, No. 3, March 1979.

6 UCB GaAs MESFET Characterization

7
Curtice GaAs MESFET Characterization

Curtice GaAs MESFET Model 451
Simulators 452
Model Parameters 454
Test Instruments 458
Measuring and Extracting 460
Extraction Algorithms 471
References 473

This chapter describes the Curtice GaAs MESFET transistor supported by HPSPICE. Descriptions of model setup, instrument connections, and model parameters are included as well as test instrument information. Procedures for extracting AC and DC model parameters from GaAs MESFET transistors using the Curtice GaAs MESFET model are also included. These model parameters describe the operating characteristics of the device under test (DUT), and can be derived from either simulated or direct measurements of the DUT.

The IC- CAP Curtice GaAs modeling module provides setups that can be used for general measurement and model extraction for GaAs technology. Two example files are provided for the Curtice GaAs MESFET transistor:

CGaas1.mdl extracts parameters for the quadratic model

CGaas2.mdl extracts parameters for the cubic model

The IC- CAP system offers the flexibility to modify any measurement or simulation specification.

The model extractions provided are also intended for general GaAs IC processes. If you have another method of extracting specific model parameters, you can do so with the Program function or by writing a function in C and linking it to the function list. Details on the Program transform and writing user-defined C language routines are explained in Chapter 9, "Using Transforms and Functions," in the IC-*CAP User's Guide*.

Curtice GaAs MESFET Model

The Curtice GaAs MESFET model is contained in the IC-CAP example files *CGaas1.mdl* and *CGaas2.mdl*. These files consist of two levels of models that are based on a model developed at RCA and implemented in many SPICE versions. In level 1 (quadratic) model (*CGaas1.mdl*), drain current is proportional to the square of the gate to source voltage multiplied by the hyperbolic tangent of the drain to source voltage [1]. Level 2 (Cubic) model (*CGaas2.mdl*), relates the drain current to the third-order polynomial of the gate and drain voltages times the hyperbolic tangent of the drain voltage [2].

The IC-CAP implementation of the model also includes voltage-dependent capacitances, gate-to-source and gate-to-drain junction diodes, and the series resistances and inductances at the gate, source, and drain.

Simulators

This model is supported only by the HPSPICE simulator in IC-CAP.

NOTE Simulators are provided as a courtesy to IC-CAP users; they are not supported by Agilent Technologies.

The default nominal temperature for SPICE3 is 25°C. To force another nominal temperature, set the *TNOM* variable to the desired value.

The HPSPICE syntax for the Curtice GaAs MESFET element statement is:

JXXXXXXX ND NG NS MNAME

where:

JXXXXXXX	indicates Curtice MESFET device name (any name that begins with J)
ND	indicates drain node number
NG	indicates gate node number
NS	indicates source node number
MNAME	indicates model name

An example of the Curtice GaAs MESFET device model call is:

```
J2 1 2 3 CMES
```

The HPSPICE syntax for the *.MODEL* definition is

.MODEL MNAME RCAY TYPE PNAME1=PVAL1 PNAME2=PVAL2 ...

where:

MNAME	indicates model name

RCAY	key word specifying a GaAs MESFET model
TYPE	indicates NJF (N-channel MESFET) or PJF (P-channel MESFET)
PNAME#	indicates Curtice GaAs MESFET parameter name
PVAL#	indicates parameter value of PNAME#

The parameter *MODEL* in the .MODEL description must be set:

MODEL = 1 indicates Curtice level 1 *quadratic* model

MODEL = 2 indicates Curtice level 2 *cubic* model

Model Parameters

Curtice GaAs MESFET model parameters are summarized in the following table. Table 51 lists the parameters with descriptions and default values. Setup attributes are listed in Table 52.

Table 50 Summary of Curtice GaAs MESFET Model Parameters

Parameter Type	Controlling Model Parameters
Inductance and Resistance	LD, LG, LS, RD, RG, RS
Diode	IS, VBI, N
Threshold	Level 1: VTO Level 2: A0, A1, A2, A3
Linear and Saturation	Level 1: BETA, LAMBDA, ALPHA Level 2: BETA, GAMMA
Capacitance and AC	CGDO, CGSO, CDS, RDSO, RIN, A5 optionally: CGD, CGS, TAU

Table 51 Curtice GaAs MESFET Model Parameters

Name	Description	Default
Inductance and Resistance Parameters		
LD	Drain Inductance. Specifies external drain inductance.	0 Henry
LG	Gate Inductance. Specifies external gate inductance.	0 Henry
LS	Source Inductance. Specifies external source inductance.	0 Henry
RD	Drain Resistance. Specifies external drain resistance.	0 Ohm
RG	Gate Resistance. Specifies external gate resistance.	0 Ohm
RS	Source Resistance. Specifies external source resistance.	0 Ohm

Table 51 Curtice GaAs MESFET Model Parameters (continued)

Name	Description	Default
Diode Parameters		
IS	Diode Reverse Saturation Current. Models gate-drain and gate-source current.	1 x 10^{-14} Amp
VBI	Gate Junction Potential. Models built-in potentials of gate-source and gate-drain regions.	0.8 Volt
N	Diode Emission Coefficient. Models emission coefficient of an ideal diode. In TECAP and some simulators this parameter is called XN.	1.0
XTI	Diode Saturation Current. Temperature Coefficient.	3.0
EG	Diode Energy Gap	1.11 EV
DC Parameters: Level 1 (Quadratic)		
ALPHA	Coefficient of VDS. It is the Vds coefficient in the tanh function of the quadratic equation.	2.0 V^{-1}
BETA	Transconductance Parameter. Defines transconductance in the saturation or linear operating regions. Same as JFET model. In TECAP and some simulators this parameter is called BETA1 for level 1, and BETA2 for level 2.	1 x 10^{-4} AÞV^{-2}
LAMBDA	Channel Length Modulation Parameter. Models the finite output conductance of a MESFET in the saturation region.	0V^{-1}
VTO	Threshold Voltage. Models gate turn-on voltage. Same as JFET model.	0V
DC Parameters: Level 2 (Cubic)		
GAMMA	Coefficient of VDS. It is the Vds coefficient in the tanh function of the cubic equation.	0.5V^{-1}
BETA	Coefficient for pinchoff. Defines change with respect to VDS In TECAP and some simulators this parameter is called BETA1 for level 1, and BETA2 for level 2.	1 x 10^{-4} A·V^{-1}
A0	0-Order Coefficient for V1 in IDS cubic equation.	1 x 10^{-2} Amp

Table 51 Curtice GaAs MESFET Model Parameters (continued)

Name	Description	Default
A1	1st-Order Coefficient for V1 in IDS cubic equation.	1×10^{-3} A·V^{-1}
A2	2nd-Order Coefficient for V1 in IDS cubic equation.	-1×10^{-3} A·V^{-2}
A3	3rd-Order Coefficient for V1 in IDS cubic equation.	-1×10^{-4} A·V^{-3}
VDSO	Value of VDS at which A0 through A3 are determined	5.0 Volt
RDSO	Internal Resistance. Drain to Source AC Leakage Path. In TECAP and some simulators this parameter is called RDS	1×10^{12} Ohm
VDSDC	VDS Bias at which RDSO, CGD and CGS are determined	0V
AC and Other Common Parameters		
A5	Proportionality Constant. Varies TAU as a function of VDS. Use TAU for constant time delay or A5 to vary delay as a function of VDS.	0 S·V^{-1}
TAU	Internal Time Delay. Constant internal time delay from drain to source.	0 Sec
VBR	Reverse Breakdown Voltage. From gate to drain.	100V
RIN	Series Resistance. In series with CGS. Used to model the change in input impedance with frequency.	0 Ohm
Piecewise Linear Current Parameters		
R1	Approximate Breakdown Resistance. R1 is the breakdown resistance from drain to gate.	0 Ohm
R2	Resistance Relating Breakdown Voltage. R2 is the resistance relating breakdown voltage to channel current.	0 Ohm
RF	Effective Value. RF if the effective value of forward-bias resistance gate to source.	0 Ohm
Constant Capacitance Parameters		
CGD	Gate to Drain Capacitance	0 Farad
CGS	Gate to Source Capacitance	0 Farad

Table 51 Curtice GaAs MESFET Model Parameters (continued)

Name	Description	Default
CDS	Drain to Source Capacitance	0 Farad
FC	Coefficient for forward-bias depletion capacitance	0.5
Non-Linear Capacitance Parameters		
CGDO	Zero bias Junction Capacitance. Non-linear Gate to Drain Capacitance at zero DC bias.	0 Farad
CGSO	Zero bias Junction Capacitance. Non-linear Gate to Source Capacitance at zero DC bias.	0 Farad

Table 52 Setup Attributes for the Curtice GaAs Model

DUT / Setup	Inputs	Outputs	Transform	Function	Extractions
ac/ s_at_f	vg, vd, vs, freq	s	extract_L_and_R	GAASAC_l_and_r	LD, LG, LS, RD, RG, RS
dc/ igvg_0v[sd]	vg, v[sd]	ig	extract	GAASDC_lev1	VBI, IS, N
			optim1	Optimize	VBI, IS, N
dc/ idvg_hi_vd	vg, vd, vs	id, ig	extract	GAASDC_cur1	Level 1: VTO Level 2: A0, A1, A2, A3
			optim	Optimize	Level 1: VTO Level 2: A0, A1, A2, A3
dc	vd, vg, vs	id, ig	extract	GAASDC_cur2	Level 1: BETA, ALPHA, LAMBDA Level 2: BETA, GAMMA
			optim	Optimize	Level 1: BETA, ALPHA, LAMBDA Level 2: BETA, GAMMA
ac	vg, vd, vs, freq	s	extract_CV	GAASC_cur	CGDO, CGSO, CDS, RDSO, RIN, A5 optionally: CGS, CGD,TAU

Test Instruments

The HP 4141, Agilent 4142, or HP 4145 can be used to derive DC model parameters from measured DC voltage and current characteristics.

The Agilent 8510, Agilent 8753, or HP 8702 (with an HP 41xx instrument) can be used to derive capacitance and inductance model parameters from S-parameter measurements.

Instrument-to-Device Connections

When the device is installed in a test fixture, verify the identity of device nodes by checking the inputs and outputs for the appropriate DUTs. Table 53 is a cross reference of the connections between the terminals of a typical MESFET device and various measurement units. These connections and measurement units are defined in *CGaas1.mdl* and *CGaas2.mdl* example files.

Input and output tables in the various setups use abbreviations D (drain), G (gate), and S (source) for the MESFET device nodes. These nodes are defined in the Circuit folder.

Measurement units (abbreviated as follows) are defined in Hardware Setup.

SMU# for DC measurement units
NWA for network analyzer units

Table 53 Instrument-to-Device Connections

DUT	Source	Gate	Drain	Comments
dc	SMU3	SMU2	SMU1	
ac	Ground	SMU2 NWA (port 1)	SMU1 NWA (port 2)	Calibrate for reference plane

Table 53 Instrument-to-Device Connections

DUT	Source	Gate	Drain	Comments

Notes:
1. DUT is the name of the DUT as specified in DUT-Setup. *Source*, *Gate*, and *Drain* are the names of the transistor terminals.
2. As an example of how to read the table, the first line indicates that the DUT named *dc* has the DC measurement instruments SMU1 connected to its drain, SMU2 connected to its gate, and SMU3 connected to its source.

Measuring and Extracting

This section general information as well as procedures for performing measurements and extractions of MESFET devices.

Measurement and Extraction Guidelines

The following guidelines are provided to help you achieve more successful model measurements and extractions.

Setting Instrument Options

Before starting a measurement, you can quickly verify instrument options settings. Save the current instrument option settings by saving the model file to *<file_name>.mdl* from the model window. Some of the Instrument Options specify instrument calibration. For the most accurate results, calibrate the instruments before taking IC-CAP measurements.

- DC measurements are generally taken with *Integration Time = Medium*.
- CV measurements in the femtofarad region usually require *High Resolution = Yes* and *Measurement Freq (kHz) = 1000*.
- When taking AC measurements with a network analyzer, several instrument settings are critical. In addition, the calibration must be performed on structures that have similar impedances as the stray parasitics of the DUT.
- Input power to the device is typically –10 to –30dBm (after port attenuation).
- Setting the averaging factor to the 2-to-4 range reduces measurement noise.
- Because IC-CAP requires the instrument to perform error correction, set *Use Internal Calibration* to *Yes*.

Experiment with the other network analyzer options to obtain the best results with specific devices.

Measuring Instruments

Ensure that the measuring instruments (specified by unit names in the input and output tables) are correctly connected to the DUT. Refer to Table 53 for a list of nodes and corresponding measurement units. The quality of the measuring equipment (instruments, cables, test fixture, transistor sockets, and probes) can influence the noise level in the measurements.

Ensure that all characteristics of the measurement stimulus and corresponding measured response are specified in the respective input and output tables.

Calibration

For some measurements the instruments or test hardware must be calibrated to remove non-device parasitics from the DUT. For MESFET devices, stray capacitance due to probe systems, bond pads, and so on should be calibrated out prior to each measurement.

For high-frequency 2-port measurements with a network analyzer, the reference plane of the instrument must be calibrated out to the DUT. IC-CAP relies on the internal calibration of the instruments for full error-corrected data. It is critical that calibration using *OPEN, SHORT, THRU,* and *50 ohm* loads be properly done.

Extracting Model Parameters

IC-CAP's extraction algorithms exist as transforms in the function list, under *Extractions*. Extraction transforms for a given setup are listed in the transform tile for the setup.

When the *extract* command is issued from the setup level, all extractions in the setup are performed in the order listed in the setup; this order is usually critical to proper extraction performance. The extractions are typically completed instantly and the newly extracted parameter values are placed in *Model Parameters*.

The configuration file supplied with IC-CAP contains the setups for two different extraction methods, and two different sets of model parameters (level 1 and 2). In general only one set of these parameters is important, and you need to perform only one of the methods in order to extract model parameters. Set the parameter MODEL to the desired number (1 or 2) before starting the extraction.

Simulating Device Response

Simulation uses model parameter values currently in the Parameters table. A SPICE deck is created and the simulation performed. The output of the SPICE simulation is then read into IC-CAP as simulated data.

HPSPICE is the only simulator fully compatible with the IC-CAP Curtice GaAs model configuration file. This simulator uses the JFET convention for calling the model. Figure 174 is an example of the circuit definition for the *Curtice GaAs MESFET* model to be used with this simulator. *JCGAAS* is the device name; *NMF1* is the model name; RCAY specifies to use the *Curtice GaAs* model; *NJF* specifies the N type FET, which is the only type supported in this model. MODEL = 1 specifies the level 1 model.

Simulations vary in the amount of time they take to complete. DC simulations generally run much faster than cv and AC simulations.

If simulated results are not as expected, use the simulation debugger (in the *Tools* menu) to examine the input and output simulation files. The output of manual simulations is not available for further processing by IC-CAP functions such as transforms and plots.

```
.SUBCKT CGAAS  1=D  2=G  3=S
JCGASS   11  22  33  NMF1
LD  1 11  1n
LG  2 22  1n
LS  3 33  1n
.MODEL  NMF1  RCAY  NJF
+ MODEL = 1
+ BETA  = 100u
+ VBI   = 0.7
+ .....
+ .....
.ENDS
```

Figure 174 Example of the circuit definition for HPSPICE

Displaying Plots

The Display Plot function displays all graphical plots defined in a setup. The currently active graphs are listed under the Plots folder in each setup. View the plots for agreement between measured and simulated data. Measured data is displayed as a solid line; simulated data is displayed as a dashed or dotted line.

Optimizing Model Parameters

Optimization of model parameters improves the agreement between measured and simulated data. An optimize transform whose *Extract Flag* is set to *Yes* is automatically called after any extraction that precedes it in the transform list.

Optimizing AC parameters can be very time consuming because of the number of SPICE simulations required.

Extraction Procedure Overview

This section describes the general procedure for extracting model parameter data from the Curtice GaAs MESFET. The procedure applies to all types of parameters. The differences between extracting one type and another lie primarily in the types of instruments, setups, and transforms used.

Parameters can be extracted from measured or simulated data. Measured data is data taken directly from instruments connected to the DUT inputs and outputs. Simulated data are

results from the simulator. Once measured and simulated data have been obtained, both data sets can be plotted and compared.

The general extraction procedure is summarized next, starting with the measurement process.

1 Install the device to test in a test fixture and connect the test instruments.

2 Ensure the test fixture, signal source and measuring instruments, and workstation are physically and logically configured for the IC-CAP system.

3 Choose **File > Open > Examples**. Select **CGaas2.mdl** and choose **OK**. Select **Open** to load the file and open the model window. Choose **OK**.

When the *CGaas2* model window opens you are ready to begin measurement and extraction operations.

4 Enter the variable name `EXTR_PAR` at the model level and enter `NMF1` as its value. This allows the extractions to find the model parameters for the model name *NMF1* within the subcircuit of the model file. This concept is covered in more detail in Chapter 9, “Circuit Modeling”.

5 Select the DUT and Setup.

6 Execute the *Measure* command.

7 Execute the *Extract* command.

8 Execute the *Simulate* command.

9 Display the results.

10 Fine tune the extracted parameters if needed by optimizing.

Parameter Measurement and Extraction

The recommended method for extracting Curtice GaAs model parameters is presented next. In this extraction, external resistances are extracted from AC data.

NOTE If AC data is not available, an alternative method (described in the section "Alternate Extraction Method" on page 467) uses the Fukui technique [3] for extracting the resistances from DC data. Use the alternative method only if AC data is not available; the recommended method produces parameters that are more precise.

The Curtice GaAs MESFET model is a 2-level model. IC-CAP supports and extracts parameters for both levels of this model. The following procedure extracts parameters for level 1 or 2 depending on the value of the parameter *MODEL*.

Parameter extractions are dependent on each other; to ensure accuracy extractions must be done in this order:

Inductance and resistance parameters (AC) External inductance and resistance parameters are extracted from an S-parameter measurement at a single bias setting. The gate of the device is strongly forward biased to make the device look like a short circuit. The *s_at_f* setup is used to take the measurements and extract the parameters LD, LG, LS, RD, RG, and RS.

Diode parameters (DC) Diode parameters VBI, IS, and N are extracted from data produced by the measurement of Id versus Vg measured at zero drain voltage, with the source floating. The *igvg_0vs* or *igvg_0vd* setup is used to make the measurements and extractions, depending on whether the gate-source or gate-drain junction is preferred.

Threshold parameters (DC) Parameters that describe the threshold characteristics are extracted using Id versus Vg measurement at a high drain voltage. The *idvg_hi_vd* setup is used for this extraction. For the level 1 model, VTO will be extracted; for the level 2 model, A0, A1, A2 and A3 will be extracted.

Linear and saturation parameters (DC) Parameters that control the linear and saturation regions of device operation are extracted using Id versus Vd measurement at different gate voltages. The *idvd_vg* setup is used for this extraction.

BETA, LAMBDA and ALPHA are extracted for the level 1 model; BETA and GAMMA are extracted for the level 2 model.

Capacitance parameters (AC) Capacitance parameters CGDO and CGSO, and AC parameters A5, CDS, RDSO and RIN are extracted from an S-parameter measurement using the *s_vs_f* setup. Measured data is corrected using the inductances and resistances extracted in the initial step; capacitance and other AC parameters are then extracted from corrected data.

By defining IC-CAP system variables *LINEAR_CGS*, *LINEAR_CGD*, and *CONSTANT_TAU* and setting their values to true, CGS, CGD, and TAU, respectively, can be extracted.

Use a network analyzer to make the next set of measurements. S-parameter measurements are highly sensitive—it is important that the instrument be properly calibrated.

1 Place the device to be measured in the test fixture.

NOTE For the *ac/s_at_f* and *s_vs_f* measurements, the SMUs connected to the network analyzer's port bias connections must correspond to the SMUs in Table 53.

2 Select the **ac/s_at_f** DUT/Setup and choose **Measure**.

3 In the ac/s_at_f DUT/Setup select **Extract** to extract the inductance and resistance parameters.

4 Select the **ac/s_vs_f** DUT/Setup and choose **Measure**. Do not extract the parameters for this setup yet.

5 Disconnect the device from the network analyzer and directly connect it to the appropriate units for DC measurements.

6 Select the **dc/ igvg_0vs** or **igvg_0vd** DUT/Setup and choose **Measure**.

7 Repeat Step 6, but choose **Extract** to extract the diode parameters.

8 Repeat Step 6, but choose **Optimize**.

9 Select the **dc/idvg_hi_vd** DUT/Setup and choose **Measure** to measure Id versus Vg at a constant Vd.

10 Select the **dc/idvd_vg** DUT/Setup and repeat the Measure and Extract steps to measure and extract the other DC parameters.

11 Repeat Step 10, but choose **Optimize**.

12 Select the **ac/s_vs_f** DUT/Setup and choose **Extract** to extract the capacitances from the data that was measured in step 4.

All model parameters are extracted and their values added to the Parameters table; they can be viewed in the Model Parameters folder.

Alternate Extraction Method

If AC data is not available, IC-CAP supports an alternate method for extracting Curtice MESFET model parameters. This procedure uses the *Fukui* technique [3]; external resistances are extracted along with the diode parameters from DC data–this differs from the recommended method. Use this method only if the AC data is not available–this alternate method produces parameters that are less precise than those of the recommended method.

Resistance and diode parameters (DC) Using DC measurements only, this procedure uses the *Fukui* algorithm to extract resistance parameters RD, RG, and RS from DC data. Diode parameters VBI, IS, and N are also extracted. The extraction requires the setups listed in the following table.

Table 54 Resistance and Diode Measurement and Extraction Setups

igvg_0vd	Ig versus Vg	Vd=0, with Source floating
idvg_low_vd	Id versus Vg	Small Vd
gvg_0vs	Ig versus Vg	Vs=0, with Drain floating

This extraction is located in the *igvg_0vd* setup. To use the Fukui algorithm, add the following inputs to the function *GAASDC_lev1*.

VG (low Vds) idvg_low_vd/vg
VD (low Vds) idvg_low_vd/vd
ID (low Vds) idvg_low_vd/id.m
VG (D Flt) igvg_0vs/vg
IG (D Flt) igvg_0vs/ig.m

Threshold Parameters ( ) Use the recommended method for measuring and extracting threshold parameters.

Linear & Saturation Parameters ( ) Use the recommended method for measuring and extracting linear & saturation parameters.

Inductance Parameters (AC) Use the recommended method for measuring and extracting inductances. The parameters LD, LG, and LS are extracted from the S-parameter measurement. In addition, the Transform also extracts and overwrites the resistance parameters.

Capacitance Parameters ( ) Use the recommended method for measuring and extracting capacitance parameters.

The alternate extraction procedure follows.

1 Connect the NWA to extract inductances and capacitances.

2 Place the device to be measured in the test fixture.

NOTE

For the *ac/s_at_f* and *s_vs_f* measurements, the SMUs connected to the network analyzer's port bias connections must correspond to the same SMUs in Table 53.

3 Select the **ac/s_at_f** DUT/Setup and choose **Measure**.

4 Repeat Step 2, but select **Extract** to extract the inductance and resistance parameters

5 Select the **ac/s_vs_f** DUT/Setup and choose **Measure**. Do not extract the parameters for this setup yet.

6 Select the **ac/s_vs_f** DUT/Setup and choose **Extract** to extract the capacitances from the data that was measured in step 3.

7 Select **dc/igvg_0vs** DUT/Setup and choose **Measure**.

8 Repeat Step 8 for dc/idvg_low_vd and dc/igvg_0vd.

9 In the dc/igvg_0vd DUT/Setup, choose **Extract** to extract the resistance and diode parameters from the measured data for the three DC Setups.

10 Repeat Step 9 but choose **Optimize**.

11 Select the **dc/idvg_hi_vd** DUT/Setup and choose **Measure** to measure Id versus Vg at a constant Vd.

12 Select the **dc/idvd_vg** DUT/Setup and repeat the Measure and Extract steps to measure and extract the other DC parameters.

13 Repeat Step 10, but choose **Optimize**.

14 Select the **dc/idvd_vg** DUT/Setup and repeat the Measure and Extract steps to measure and extract the other DC parameters.

15 Repeat Step 10, but choose **Optimize.**

Simulating

To simulate any individual setup, choose Simulate with that setup active. Simulations can be performed in any order after all of the model parameters have been extracted. For more information on simulation, refer to Chapter 6, "Simulating," in the *IC-CAP User's Guide*.

Displaying Plots

To display plots of measured and simulated data issue the *Display Plots* command from a DUT to display the plots for all setups contained in that DUT.

Viewing plots is an ideal way to compare measured and simulated data to determine if further optimization would be useful. For more information on displaying plots, refer to Chapter 10, "Printing and Plotting," in the *IC-CAP User's Guide*.

Optimizing

The optimization operation uses a numerical approach to minimizing errors between measured and simulated data. As with the other IC-CAP commands, optimization can be performed at either the DUT or setup level. Optimization is more commonly performed from setups—optimization for all setups under a DUT is rarely required.

Optimization is typically interactive in nature, with the best results obtained when you specify the characteristics of the desired results.

For more information on optimization, refer to Chapter 7, "Optimizing," in the *IC-CAP User's Guide*.

Extraction Algorithms

This section describes the extraction algorithms used for inductance, capacitance, DC, and series resistance parameters of the Curtice GaAs MESFET. Setups for the Curtice MESFET model are similar to the UCB GaAs MESFET model and their extraction algorithms are similar. The Curtice model has more model parameters than the UCB model.

Inductance and Resistance Extraction

Inductors and resistors that are in series with each terminal of the MESFET are measured with a network analyzer at a high frequency and with the gate strongly forward-biased. Under these conditions the AC model for each terminal is reduced to a series R-L circuit. Inductors LD, LG, and LS and resistors RD, RG, and RS are extracted simultaneously from the measured impedance at the gate and drain ports.

DC Parameter Extractions

DC extractions are separated between the diode and forward active controlling parameters. Diode parameters VBI, IS, and N are extracted using a method similar to the method used for a silicon diode. The diode is swept over a forward bias and linear least-squares curve fits will produce the built-in potential and forward conduction properties.

The forward active region is extracted from measurements of the MESFET in both the linear and saturated modes of operation.

- If the MODEL parameter is set to 1, BETA, VTO, ALPHA, and LAMBDA are extracted.
- If the MODEL parameter is set to 2, A0, A1, A2, A3, BETA, and GAMMA are extracted.

(Note that for MODEL=1, BETA is a transconductance and for MODEL=2, BETA is a coefficient for pinchoff.)

Capacitance Parameter Extractions

The capacitance of the gate-to-drain CGDO and gate-to-source CGSO is measured with S-parameters over typical operating frequencies. Because the inductances and series resistance are already known, these capacitances can be extracted from the corrected impedances measured at the gate and drain ports.

References

1 Walter Curtice. *A MESFET Model For Use in the Design of GaAs Integrated Circuits*, IEEE Transactions on Microwave Theory & Techniques, Vol. MTT-28, No. 5, May 1980.

2 Walter Curtice and M. Ettenberg. *A Nonlinear GaAs FET Model for Use in the Design of Output Circuits for Power Amplifiers,* IEEE Transactions on Microwave Theory & Techniques, Vol. MTT-33, No. 12, December 1985.

3 H. Fukui. *Determination of the Basic Device Parameters of a GaAs MESFET*, The Bell System Technical Journal, Vol. 58, No. 3, March 1979.

7 Curtice GaAs MESFET Characterization

8
MOS Model 9 Characterization

MOS Model 9 Model 477
The MM9 Model File 479
Parameter Extraction 511
Optimizing 516
The JUNCAP Model 519
References 536

MOS Model 9, developed by Philips, is a compact model for circuit simulation, suitable for both digital and analog applications. It provides the following features:

- Non-uniform doping effect on V_{TH}
- Mobility reduction due to vertical field
- V_{bs} influence on mobility reduction
- Velocity saturation
- Channel length modulation
- Subthreshold conduction
- DIBL/Static-feedback
- Substrate current
- Parameter scaling with respect to W, L, and temperature
- Based on single-equation I-V and Q-V formulations
- Continuous g_m, g_m/I_d and g_{ds} behavior in the weak to strong inversion and linear to saturation transition regions

This implementation is intended only for enhancement mode MOSFETs. Although MOS Model 9 also has applications for depletion mode devices, this implementation does not support

this option. It is intended to work in the absence of a circuit simulator with MOS Model 9 being available to IC-CAP. Thus, the MOS Model 9 equations are implemented with C routines that are linked directly to the IC-CAP executable.

The suitability and accuracy for DC, AC and statistical applications have been demonstrated by Philips in several publications (see References [1], [2], and [3] at the end of this chapter).

MOS Model 9 Model

MOS Model 9 uses two IC-CAP models: *mm9* and *mm9_tempx*. Both of these models are stored in the file *mm9.mdl*. When saving in the Main window, ensure both model definitions are kept.

- *mm9* is the main model definition file and contains the templates for measurements and extraction.
- *mm9_tempx* is the template file for data that will be measured at non-nominal temperature. The most important aspect of this file is that the MM9 parameter values are set to the values in the model *MM9*.

The primary method of model evaluation relies on the function *MM9*, which appears in the Function Group MM9. This function requires the inputs VD, VG, VS, and VB, which are arrays that give the drain, gate, source, and bulk voltages, respectively. It also requires the parameter *Output*, which controls the current returned by the function and is defined by one of the following options:

- D to return drain current
- S to return source current
- B to return bulk (avalanche) current

The calculations performed by this function are also influenced by two variables (MODLEVEL and EQNTYPE). These quantities and their influence are shown in Table 56.

The following figure illustrates the overall structure of the model.

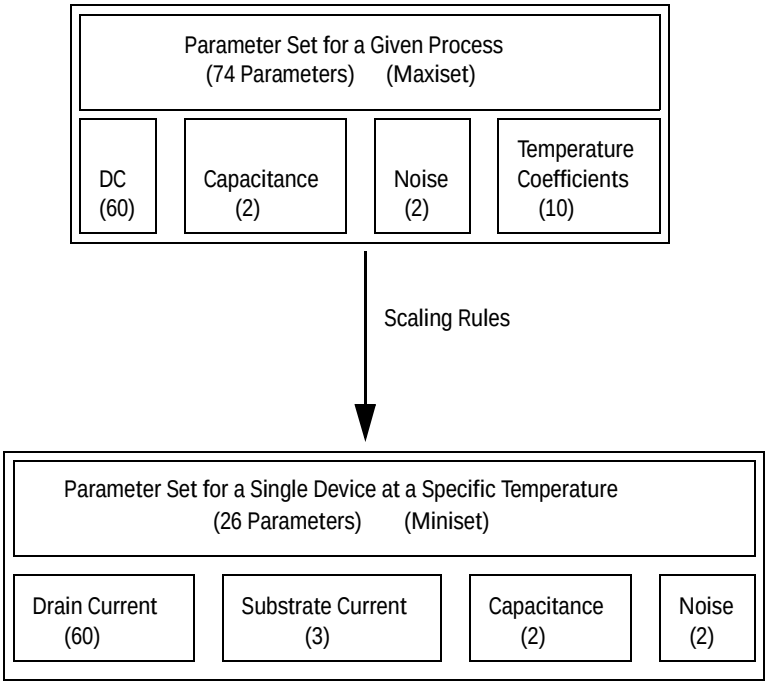


Figure 175 Overall Structure of MOS Model 9

The MM9 Model File

This section describes the MOS Model 9 model parameters, model variables, DUT/setup details, and macros.

Model Parameters

The following table describes the MOS Model 9 parameters.

Table 55 MOS Model 9 Parameters

Parameter	Description	Default
LER	Effective channel length of the reference transistor.	501.3n
WER	Effective channel width of the reference transistor.	9.787u
LVAR	Difference between the actual and the programmed poly-silicon gate length.	-198.7n
LAP	Effective channel length reduction per side due to the lateral diffusion of the source/drain dopant ions.	0.000
WVAR	Difference between the actual and the programmed field-oxide opening.	-212.7n
WOT	Effective channel width reduction per side due to the lateral diffusion of the channel-stop dopant ions.	0.000
TR	Temperature at which the parameters have been determined.	27.00
VTOR	Threshold voltage at zero back-bias.	810.2m
STVTO	Coefficient of the temperature dependence of VTO.	-1.508m
SLVTO	Coefficient of the length dependence of VTO.	18.95n
SL2VTO	Second coefficient of the length dependence of VTO.	-15.09f
SWVTO	Coefficient of the width dependence of VTO.	55.35n
KOR	Low-back-bias body factor.	610.0m
SLKO	Coefficient of the length dependence of KO.	-76.45n
SWKO	Coefficient of the width dependence of KO.	55.79n

Table 55 MOS Model 9 Parameters

Parameter	Description	Default
KR	High-back-bias body factor.	170.5m
SLK	Coefficient of the length dependence of K.	-293.1n
SWK	Coefficient of the width dependence of K.	185.5n
VSBXR	Transition voltage for the dual-k factor model.	1.926
SLVSBX	Coefficient of the length dependence of VSBX.	443.2n
SWVSBX	Coefficient of the width dependence of VSBX.	-349.8n
BETSQ	Gain factor.	155.9u
ETABET	Exponent of the temperature dependence of the gain factor.	1.655
THE1R	Coefficient of the mobility due to the gate induced field.	306.3m
STTHE1R	Coefficient of the temperature dependence of THE1.	-613.8u
SLTHE1R	Coefficient of the length dependence of THE1.	66.10n
STLTHE1	Coefficient of the temperature dependence of the length dependence of THE1.	-95.78p
SWTHE1	Coefficient of the width dependence of THE1.	-44.23n
THE2R	Coefficient of the mobility due to the back-bias.	43.49m
STTHE2R	Coefficient of the temperature dependence of THE2.	97.75u
SLTHE2R	Coefficient of the length dependence of THE2.	-56.97n
STLTHE2	Coefficient of the temperature dependence of the length dependence of THE2.	-13.64p
SWTHE2	Coefficient of the width dependence of THE2.	19.14n
THE3R	Coefficient of the mobility due to the lateral field.	264.4m
STTHE3R	Coefficient of the temperature dependence of THE3.	8.227u
SLTHE3R	Coefficient of the length dependence of THE3.	135.8n
STLTHE3	Coefficient of the temperature dependence of the length dependence of THE3.	-509.4p

Table 55 MOS Model 9 Parameters

Parameter	Description	Default
SWTHE3	Coefficient of the width dependence of THE3.	-20.09n
GAM1R	Coefficient for the drain induced threshold shift for large gate drive.	65.48m
SLGAM1	Coefficient of the length dependence of GAM1.	28.22n
SWGAM1	Coefficient of the width dependence of GAM1.	-9.967n
ETADSR	Exponent of the **VDS** dependence of GAM1.	600.0m
ALPR	Factor of the channel-length modulation.	6.248m
ETAALP	Exponent of length dependence of ALP.	0.000
SLALP	Coefficient of the length dependence of ALP.	0.000
SWALP	Coefficient of the width dependence of ALP.	4.761n
VPR	Characteristic voltage of channel length modulation.	443.5m
GAMOOR	Coefficient of the drain induced threshold shift at zero gate drive.	20.40m
SLGAMOO	Coefficient of the length dependence of GAMO.	5.295f
ETAGAMR	Exponent of the back-bias dependence of GAMO.	2.000
MOR	Factor of the subthreshold slope.	536.6m
STMO	Coefficient of the temperature dependence of MO.	470.4u
SLMO	Coefficient of the length dependence of MO.	164.2u
ETAMR	Exponent of the back-bias dependence of M.	2.000
ZET1R	Weak-inversion correction factor.	1.815
ETAZET	Exponent of length dependence of ZET1.	500.0m
SLZET1	Coefficient of the length dependence of ZET1.	-1.413m
VSBTR	Limiting voltage of the VSB dependence of M and GAMO.	15.97
SLVSBT	Coefficient of the length dependence of VSBT.	10.12u
A1R	Factor of the weak-avalanche current.	61.47

Table 55 MOS Model 9 Parameters

Parameter	Description	Default
STA1	Coefficient of the temperature dependence of A1.	50.07m
SLA1	Coefficient of the length dependence of A1.	-907.0n
SWA1	Coefficient of the width dependence of A1.	-7.211u
A2R	Exponent of the weak-avalanche current.	31.48
SLA2	Coefficient of the length dependence of A2.	-877.5n
SWA2	Coefficient of the width dependence of A2.	-923.4n
A3R	Factor of the drain-source voltage above which weak-avalanche occurs.	755.6m
SLA3	Coefficient of the length dependence of A3.	-114.4n
SWA3	Coefficient of the width dependence of A3.	12.17n
TOX	Thickness of the oxide layer.	15.00n
COL	Gate overlap per unit channel width.	100.0p
NTR	Coefficient of the thermal noise.	0.000
NFR	Coefficient of the flicker noise.	0.000

Model Variables

The following table describes the MOS Model 9 model variables.

Table 56 MM9 Variables

Variable Name	Description	Default Value
VP_large	VP of large device in dataset	4.210
L_large	Length of large device in dataset	10.00u
SETUP_LIST_SIZE	Default number of visible setups	1
MACRO_LIST_SIZE	Default number of visible macros	16
VAR_ROW_SIZE	Default number of visible variables	22
PARAM_ROW_SIZE	Default number of visible parameters	22

Table 56 MM9 Variables

Variable Name	Description	Default Value
VSUP	Maximum bias voltage	5.000
NUMDUT	Number of devices in dataset	14.00
DUT	Present device being measured/extracted	2
COMGATE	Scanner pin connected to common device gate	20.00
COMSOURCE	Scanner pin connected to common device source	26.00
COMBULK	Scanner pin connected to common device bulk	18.00
MATADD	Address of scanner as used in SWM_init statement	22.00
MATNAME	Name of scanner as used in SWM_init statement	HP4085B
DUT_LARGE	Index number for the DUT considered to be large	1.000
YLOW	Low bound for drain current optimization	500.0f
YHIGH	High bound for drain current optimization	1
KFACTOR	Choice of 1 or 2 K-factor model	2.000
YLOW_SUB	Low bound for substrate current optimization	-1
YHIGH_SUB	High bound for substrate current optimization	-5E-13
LIN_VGSSTEP	Vgs step size for linear region curves	100.0m
VBS1	Vbs bias used for saturation and subthreshold sweeps	0.000
VBS2	Vbs bias used for saturation and subthreshold sweeps	2.000

Table 56 MM9 Variables

Variable Name	Description	Default Value
VBS3	Vbs bias used for saturation and subthreshold sweeps	5.000
SAT_DELVGS	First saturation region curve in idvg1 is measured for VGS = TYPE * (VTH + SAT_DELVGS)	100.0m
SAT_VGS2	Vgs value for saturation region curves	2.000
SAT_VGS3	Vgs value for saturation region curves	3.500
SAT_VGS4	Vgs value for saturation region curves	5.000
SAT_VDSSTEP	Vds step size for saturation region curves	100.0m
SVT_DELVGS1	For the subthreshold curves Vgs is varied from	600.0m
SVT_DELVGS2	TYPE*(VTH-SVT_DELVGS1) to TYPE*(VTH-SVT_DELVGS2)	300.0m
SVT_VGSSTEP	Vgs step size for subthreshold region curves	50.00m
SUB_VDS1	Vds value for substrate current curves	4.000
SUB_VDS2	Vds value for substrate current curves	4.500
SUB_VDS3	Vds value for substrate current curves	5.000
SUB_VGSSTEP	Vgs step size for substrate current curves	100.0m
SVT_VDS1	Vds value for subthreshold curves	1.000
SVT_VDS2	Vds value for subthreshold curves	3.000
SVT_VDS3	Vds value for subthreshold curves	5.000
LIN_VDS	Vds for linear region curves	100.0m
NUMLPLOT	Array size for the data in extract/par_vs_L	7.000
NUMWPLOT	Array size for the data in extract/par_vs_W	5.000

Table 56 MM9 Variables

Variable Name	Description	Default Value
NUMRPLOT	Array size for the data in extract/par_vs_R	3.000
IMIN	Low current limit used for determining optimization targets and the minimum current predicted by MM9	500.0f
EQNTYPE	Allows equation simplification for linear parameter extraction: 0 = Use normal parameter extraction equations. 1 = Use a simplification to help linear region extraction. 2 = Use the extended equations that would be implemented in a circuit simulator.	0
MODLEVEL	Selects equation and parameter set for miniset, maxiset, single temperature or all temperature extraction: 0 = Use the miniset parameters to evaluate the currents. These miniset parameters are read from the variable table of the DUT from which MM9 is invoked. 1 = Use the maxiset parameters and the full scaling rules but assuming operation at nominal temperature. The maxiset parameters are read from the model parameter list. 2 = Use the full geometry and temperature scaling rules (i.e., the normal model equation). The model parameters are read from the model parameter list. 3 = Use the full geometry scaling rules	1
TYPE	Device type: 1 for NMOS, -1 for PMOS	1.000
TEMP	Measurement temperature	21.00
MULTDUT	Indicates if there are multiple transistors connected in parallel	N

Table 56 MM9 Variables

Variable Name	Description	Default Value
PROBETYPE	Indicates how the devices are to be connected: M : manually; A: automatically with a scanner	M
NUMTEMP	Number of temperatures at which the devices will be measured for temperature parameter extraction	2.000
GEOMFILE	Name of system file in which the miniset parameters will be temporarily stored	mm9_geompars
TEMPFILE	Name of system file in which the temperature specific parameters will be temporarily stored	mm9_temppars
LIN_NUMVBS	Number of curves measured in the linear region	6.000
DISPLAYPLOTS	Automatically displays plots when measuring or optimizing. *Hint:* For a small number of devices, such as two, you may want to set this variable to Y. When measuring or optimizing three or more, set this variable to *N*.	N
DATASOURCE	Enables you to generate *measured* data from the model code if measured data is not available. To do this, set this variable to S and execute one of the measure macros. When measuring real data, this variable must be set to *M*.	M
SWAPDIRECTION	Help variable used during the setup of the non-nominal temperature models	1
TA_SWAP	Help variable used to set temperature	100
NUMTPLOT	Array size for the data in extract/par_vs_T	3.000

Table 56 MM9 Variables

Variable Name	Description	Default Value
GDSMIN	Low GDS limit used for determining optimization targets	1.000p
VBSTOP	Last value of Vbs for linear region	5.0
LIMIT_FLAG	Indicates if one of the parameters is at its allowed limit	0
ERROR	Used to indicate an error with the quick extraction routines for the linear, subthreshold or saturation regions	0
THE3_STORE	Temporary store for THE3	
RECALC	Indicates whether a quick extraction function should do a measurement or use existing data	0
Linear Region Variables for Quick Extraction		
VSBREF	A reference value of Vsb to set the threshold voltage to at the end of the quick extraction routines.	0
VT_RANGE	The maximum expected change in threshold voltage between successive iterations. If the change in threshold voltage exceeds this value, an error occurs.	3
K_MODEL	Choice of K-factor model 1: a single K-factor is used 2: the dual K-factor model is used	2
DOBODY	Control variable for body-effect parameters 0: no body-effect parameters are extracted 1: body-effect parameters are extracted	1
VGATE1	First gate overdrive voltage	0.6
VGATE2	Second gate overdrive voltage	1.5
VGATE3	Third gate overdrive voltage	3.5

Table 56 MM9 Variables

Variable Name	Description	Default Value
VSB11	1st Vsb	0
VSB12	2nd Vsb	0.3
VSB21	3rd Vsb	4
VSB22	4th Vsb	5
VTHMAX	The maximum absolute value of threshold voltage anticipated for the device under test.	-0.15
VDSPRG	The drain voltage to be used during linear region extractions.	0.1
Subthreshold Region Variables for Quick Extraction		
VDSSTH1	1st Vds	1
VDSSTH2	2nd Vds	5
VGATST1	Offset from threshold voltage of 1st Vgs bias	-0.15
VGATST2	Offset from threshold voltage of 2nd Vgs bias	-0.2
VSBSTH1	1st Vbs	0
VSBSTH2	2nd Vbs	5
Saturation Region Variables for Quick Extraction		
NUMIDS	Number of points chosen to optimize with respect to ids.	3
NUMGDS	Number of points chosen to optimize with respect to gds	3
VSBSAT	Vbs for saturation measurements	0
DVDGDS	The increment in drain voltage to be used when measuring output conductance	0.05
Weak Avalanche Variables for Quick Extraction		

Table 56 MM9 Variables

Variable Name	Description	Default Value
VSBWA	Vbs for weak avalanche measurements	0
VGSWA1	Offset from threshold voltage of 1st Vgs	0.75
VGSWA2	Offset from threshold voltage of 2nd Vgs	0.5
VGSWA3	Offset from threshold voltage of 3rd Vgs	1.5
VDSWA1	1st Vds	5
VDSWA2	2nd Vds	6.5
QTRANS_NAME	Holds the name of the transform in quick_ext/store, which can be used to set the variables associated with quick extraction	quick_extraction_setup
KO_INIT	Initial value for KO	0.8
K_INIT	Initial value for K	0.2
VSBX_INIT	Initial value for VSBX	1.5
GAMOO_INIT	Initial value for GAMOO	0.01
MO_INIT	Initial value for MO	0.5
ZET1_INIT	Initial value for ZET1	1
VP_INIT	Initial value for VP	1.5
ALP_INIT	Initial value for ALP	0.01
THE3_INIT	Initial value for THE3	0
GAM1_INIT	Initial value for GAM1	0.01

The extract DUT

The *extract* device contains all of the sequences used for the parameter optimizations and much of the setup information.

extract/devices holds setup information that has the form of an array. The input *index* is used to establish the size of the various arrays. Its size in turn is controlled by the variable *NUMDUT* from the model variable table. Note that the DUTs are labeled from 1 to the number of devices but the arrays holding the DUT information begin with index 0. The outputs for this setup are shown next.

width	Holds the widths of the devices to be measured
length	Holds the lengths of the devices to be measured
mult	Holds the values for MULT for each device, that is, the number of similar structures connected in parallel.
drain	Holds the matrix pin numbers connected to the drains (if a switching matrix is being used)
gate	The matrix pin numbers connected to the gates
source	The matrix pin numbers connected to the sources
bulk	The matrix pin numbers connected to the bulks
dotemp	An array that indicates if the devices are to be measured at temperature. If the value of dotemp for any device is set to 1, then this device will be measured at temperature.

The *devices* setup contains the following transforms:

connect calls *SWM_init* and *Connect* to connect the matrix for a particular device or prompts you to connect the device. It uses information contained in the outputs described for the setup *devices*.

dummy is an empty (apart from comments) PEL transform. It was found that when a variable that affects the array size in any setup (NUMDUT in this case) is changed from a C transform, then a call to a dummy transform is necessary to force IC-CAP to re-establish the proper array dimensions before attempting to write to these arrays.

extract/single_ext contains the sequences for extracting the miniset parameters. The variable table of this setup contains a list of MIN and MAX values for use in the optimization steps. It is easier to modify the optimization limits from such a variable table rather than from the individual optimization transforms.

For the extraction of a miniset for any particular DUT, this setup is first copied into the appropriate DUT. The optimizations then operate on the miniset variables local to that DUT.

The *single_ext* setup contains the following transforms:

full_extract is the controlling PEL for miniset extraction. For more information, refer to the discussion on *full_extract* in the section "Optimization Transforms and Macros" on page 516.

par_init initializes parameter (local variables in fact) values at the beginning of *miniset* extraction.

lin_opt1 is an optimization call for linear region fitting at Vbs = 0 for the parameters BET, THE1, and VTO

lin_opt2 is an optimization call for linear region fitting for all Vbs for the parameters KO, THE2, VSBX, and K. This transform is used for the case of the 2 k-factor model.

lin_opt3 is an optimization call for linear region fitting for all Vbs for the parameters KO and THE2. This transform is used for the case of the 1 k-factor model.

subvt_opt1 is an optimization call for subthreshold region fitting at Vbs = 0 for the parameters GAMOO, MO, and ZET1.

normal_gds_opt1 is an optimization call for gds fitting at Vbs = 0 for the *normal devices* and for the parameters GAM1 and ALP

large_gds_opt1 calls an optimization sequence for gds for the special case of the device with the largest length. This sequence in turn calls *vp_opt* and *alp_opt*.

vp_opt is an optimization call for gds fitting at Vbs = 0 for the parameter VP.

alp_opt is an optimization call for gds fitting at Vbs = 0 for the parameter ALP

set_VP sets the VP of any device by scaling of the VP of the large device.

set_VP_large sets a model level variable *VP_large* to hold the VP of the large device.

ids_opt1 is an optimization call for ids fitting at Vbs = 0 for the parameter THE3.

isub_opt1 is an optimization call for avalanche current fitting for Vbs = 0 for the parameters A1, A2, and A3.

subvt_opt2 is an optimization call for subthreshold region fitting for all Vbs for the parameter VSBT.

limit_check is called at the end of each miniset optimization to check the parameters with respect to the miniset limits. It is used by the macros *extract_one_miniset* and *extract_all_minisets*.

extract / scaled_ext contains the optimization sequences necessary for scaled (maxiset) extraction at the nominal temperature. The variable table of this setup contains the parameter MIN and MAX limits that will be used during optimization.

The *scaled_ext* setup contains the following transforms:

sim_all cause the currents in all the DUTs at the nominal temperature to be resimulated (i.e., evaluated with the MM9 C transform).

sca_opt controls the sequence for maxiset optimizations. For more information, refer to the discussion on *sca_opt* in the section "Optimization Transforms and Macros" on page 516.

read_sca_opt_files reads the definitions of the optimization transforms from the UNIX file system. When you execute SETUP, the number of devices may change and the optimization tables for maxiset extraction need to be rebuilt. This is performed by the C transform SETUP which writes the new optimization definitions to the file system. This transform then reads these new definitions back into IC-CAP.

sca_lin_opt1 is an optimization call for linear region fitting at Vbs = 0 for the parameters VTOR, SLVTO, SL2VTO, SWVTO, BETSQ, THE1R, SLTHE1R, and SWTHE1.

sca_lin_opt2 is an optimization call for linear region fitting at all Vbs for the parameters THE2R, SLTHE2R, SWTHE2, KOR, SLKO, SWKO, KR, SLK, SWK, VSBXR, SLVSBX, and SWVSBX. This sequence is used for the 2 k-factor model option.

sca_lin_opt3 is an optimization call for linear region fitting at all Vbs for the parameters THE2R, SLTHE2R, SWTHE2, KOR, SLKO, and SWKO. This sequence is used for the 1 k-factor model option.

sca_subvt_opt1 is an optimization call for subthreshold optimization at Vbs = 0 for the parameters GAMOOR, SLGAMOO, MOR, SLMO, ZET1R and SLZET1.

sca_gds_opt1 is an optimization call for gds fitting for Vbs = 0 for the parameters GAM1R, SLGAM1, SWGAM1, ALPR, SLALP, SWALP and VPR.

sca_ids_opt1 is an optimization call for ids fitting for Vbs = 0 for the parameters THE3R, SLTHE3R, and SWTHE3.

sca_isub_opt1 is an optimization call for substrate (avalanche) current fitting at Vbs = 0 for the parameters A1R, SLA1, SWA1, A2R, SLA2, SWA2, A3R, SLA3 and SWA3.

sca_opt_subvt2 is an optimization call for subthreshold current fitting for all Vbs for the parameters VSBTR and SLVSBT.

sca_limit_check is called at the end of a maxiset extraction or optimization to check the parameters with respect to the maxiset limits. It is used by the macros *extract_maxiset* and *optimize_maxiset*.

extract/single_temp_extract contains the optimization sequences necessary for the extraction of the temperature sensitive parameters at a single temperature. It will be copied into each model that exists for a non-nominal temperature and the extraction sequences in this setup will therefore modify the variables of the model in which this setup occurs.

At a model level, the variables that represent the temperature-sensitive parameters are xVTOR, xBETSQ, xTHE1R, xSLTHE1R, xTHE2R, xSLTHE2R, xTHE3R, xSLTHE3R,

xMOR and xAIR. The variable table of *single_temp_extract* contains the upper and lower bounds that will be used during the optimization sequences.

The *single_temp_extract* setup contains the following transforms:

temp_par_init initializes the temperature-sensitive parameters at any temperature to their value at the nominal temperature.

select_single_temp_model sets MODLEVEL and EQNTYPE so that the single temperature option of the MM9 transform will be used. That is to say, where most parameters are read from the Parameters table and full geometry scaling is used, but where the values for the temperature-dependent parameters are read from the variable table of the model that has measurements at a non-nominal temperature.

swapdata is used to transfer setup information (mainly bias voltages and temperatures) from the *MM9* model to any model containing temperature data.

single_temp_opt controls the optimization sequence for temperature optimizations at one temperature. For more information, refer to the discussion on *single_temp_opt* in the section "Optimization Transforms and Macros" on page 516.

single_temp_lin_opt1 is an optimization call to fit linear region data at Vbs = 0 for all the devices at a particular non-nominal temperature.

The variables optimized are xVTOR, xBETSQ, xTHE1R, and xSLTHE1R.

single_temp_lin_opt2 is an optimization call to fit linear region data for all Vbs for all the devices at a particular non-nominal temperature.

The variables optimized are xTHE2R and xSLTHE2R.

single_temp_subvt_opt1 is an optimization call to fit subthreshold data for Vbs = 0 for all the devices at a single non-nominal temperature. The variable optimized is xMOR.

single_temp_ids_opt1 is an optimization call to fit ids at Vbs = 0 for all the devices at a single non-nominal temperature. The variables optimized are xTHE3R and xSLTHE3R.

single_temp_isub_opt1 is an optimization call to fit the avalanche current at Vbs = 0 for all the devices at a single non-nominal temperature. The variable optimized is xA1R.

single_temp_limit_check is called at the end of the parameter optimization for a single non-nominal temperature to check the single temperature parameters with respect to their limits. It is used by the macro *optimize_at_one_temperature*.

extract/all_temp_extract contains the extraction sequences needed for optimization of the temperature coefficients of MOS Model 9 for all the devices measured at all non-nominal temperatures. These parameters are ETABET, STVTO, STTHE1R, STLTHE1, STTHE2R, STLTHE2, STMO, STTHE3R, STLTHE3, and STA1.

The variable table of this setup contains the MIN and MAX limits that are to be used for these parameters during optimization.

There is one input defined in this setup (*index*) that is used to set up the array size for the output *temp*. This output holds the list of non-nominal temperatures at which you want measurements to be performed. This array will be updated whenever you execute the SETUP macro.

The *all_temp_extract* setup contains the following transforms:

dummy is an empty (except for comments) PEL that is used to re-establish the array size in this setup when the variable *NUMTEMP* changes.

read_all_temp_opt_files forces the optimization tables to be rebuilt and read from the file system whenever SETUP macro is run.

all_temp_lin_opt1 fits the linear region data at Vbs = 0 by optimizing the parameters and ETABET, STVTO, STTHE1R, STLTHE1.

all_temp_lin_opt2 fits the linear region data at all Vbs by optimizing the parameters STTHE2R and STLTHE2

all_temp_subvt_opt1 fits the subthreshold region data at Vbs = 0 by optimizing the parameter STMO.

all_temp_ids_opt1 fits the ids (saturation) data at Vbs = 0 by optimizing the parameters STTHE3R and STLTHE3.

all_temp_isub_opt1 fits the avalanche current data at Vbs = 0 by optimizing the parameter STA1.

all_temp_limit_check is used to check the overall temperature parameters with respect to their limits. It is used by the macros *extract_temperature_coefficients* and *optimize_temperature_coefficients*.

extract / par_vs_L, par_vs_W, par_vs_R, and **par_vs_T** are used to illustrate the geometry (L, W, R) and temperature (T) scaling. The *par_vs_L*, *par_vs_W*, and *par_vs_R* setups store graphs of the miniset parameters A1, A2, A3, ALP, GAM1, GAMOO, K, KO, MO, THE1, THE2, THE3, VP, VSBT, VSBX, VTO, ZET1 and BET vs. 1/Leff, 1/Weff, or 1/Reff where Reff is a dimension number associated with transistors that do not lie on the standard length and width arrays.

The parameters in these 3 setups are initially created by the C transform MM9_GEOMSCAL, which extracts the geometry scaling coefficients (the maxiset model). In any of these plots, the variables with extension *.m* (e.g., *VTO.m*) represent the values of the miniset parameters as extracted for an individual device.

The variables with extension *.s* (e.g., *VTO.s*) represent the miniset value predicted by using the scaled model. Because the scaled model can be optimized, these values can be recalculated for the new scaling coefficients by a call to the C transform MM9_GEOMPAR.

The variables with suffix *_lsq* are used to hold the initial *fits* to the miniset parameters just after the least-squares fitting in MM9_GEOMSCAL.

The *par_vs_T* setup shows the variation of the temperature-sensitive parameters VTOR, BETSQ, THE1R, SLTHE1R, THE2R, SLTHE2R, THE3R, SLTHE3R, MOR and A1R with temperature and their fitting with the temperature scaling rules.

In these plots, the extension *.m* indicates the parameter values extracted at a single temperature, while the extension *.s* indicates the predicted value of the parameter using the temperature coefficients of the current model set (assuming the plots have been updated with a call to the C transform MM9_TEMPPAR) and the suffix *_lsq* indicates the fits that were obtained by the temperature coefficients obtained from the least-squares extraction transform *MM9_TEMPSCAL*.

extract/par_vs_L2 and **par_vs_W2** enable parameter versus length plots for a user-specified width and parameter versus width plots for a user-specified length to be generated. This is useful if the device set includes more than one "L-array" and more than one "W-array."

The quick_ext DUT

The *quick_ext* device contains the measurement templates and the transforms used for quick extraction of the miniset parameters of MOS Model 9. The DUT variables are used to store the current values of the miniset parameters as they are being extracted.

quick_ext/lin_quick_ext is used during the extraction of the linear region parameters. It contains input definitions for the bias voltages vd, vg, vs, and vb, as well as the definition for the current to be measured, id. The *lin_quick_ext* setup contains the following inputs and outputs:

vd	A constant value set by the variable VDS
vg	A list with three voltages set by the variables: VGS0, VGS1, and VGS2
vs	A constant value of 0V
vb	A constant value set by the variable VBS
id	The current output from the vd terminal

The variables VDS, VGS0, VGS1, VGS2 and VBS are setup variables and are set automatically by the function MM9_LIN_EXT.

The *lin_quick_ext* setup contains the following transforms:

mm9_ids calls the MM9 transform for current simulation.

copy_ids allows current to be copied from mm9_ids to id.m.

set_dimensions sets the dimension information in the *quick_ext* DUT from the information in the *extract/devices* arrays.

linear_extract calls the linear region extraction functions.

quick_measure used by MM9_LIN_EXT to initiate measurements. If the variable DATASOURCE is set to M, then real measurements are to be performed. If not, then it is assumed that measurements are being made using an ideal miniset. This causes a little confusion because the quick extraction changes the miniset parameters as it proceeds. Thus the ideal miniset parameters and the quick extraction miniset parameters have to be used appropriately. Some transforms in the setup *store* are used to achieve this.

par_init_quick_ext sets initial values of ETAGAM and ETAM.

quick_ext / svt_quick_ext used during the extraction of the subthreshold region parameters. It contains input definitions for the bias voltages vd, vg, vs and vb, as well as the definition for the current to be measured, id. The *svt_quick_ext* setup contains the following inputs and outputs:

vd	A constant value set by the variable VD
vg	A constant value set by the variable VG
vs	A constant value set by the variable VS
vb	A constant value set by the variable VB
id	The current output from the vd terminal

The variables VD, VG, VS, and VB are setup variables and are set automatically by the function MM9_STH_EXT.

The *svt_quick_ext* setup contains the following transforms:

mm9_ids calls the MM9 transform for current simulation.

copy_ids allows current to be copied from mm9_ids to id.m.

subvt_extract calls the subthreshold region extraction functions.

quick_measure used by MM9_STH_EXT to initiate subthreshold region measurements. Its functionality is the same as that of *quick_measure* in *lin_quick_ext*.

quick_ext / sat_quick_ext used during the extraction of the saturation (including output conductance) parameters. It contains input definitions for the bias voltages vd, vg, vs, and vb, as well as the definition for the current to be measured, id. The *sat_quick_ext* setup contains the following inputs and outputs:

vd	A constant value set by the variable VD
vg	A constant value set by the variable VG
vs	A constant value set by the variable VS
vb	A constant value set by the variable VB
id	The current output from the vd terminal

The variables VD, VG, VS, and VB are setup variables and are set automatically by the function MM9_SAT_EXT.

The *sat_quick_ext* setup contains the following transforms:

mm9_ids calls the MM9 transform for current simulation.

copy_ids allows current to be copied from mm9_ids to id.m.

saturation_extract calls the saturation region extraction functions.

quick_measure used by MM9_SAT_EXT to initiate saturation region measurements. Its functionality is the same as that of *quick_measure* in *lin_quick_ext*.

quick_ext/weav_quick_ext used during the extraction of the weak avalanche parameters. It contains input definitions for the bias voltages vd, vg, vs, and vb, as well as the definitions for the current to be measured, id and ib. The *weav_quick_ext* setup contains the following inputs and outputs:

vd	A constant value set by the variable VD
vg	A constant value set by the variable VG
vs	A constant value set by the variable VS
vb	A constant value set by the variable VB
id	The current output from the vd terminal
ib	The current output from the vb terminal

The variables VD, VG, VS, and VB are setup variables and are set automatically by the function MM9_WEAVAL_EXT.

The *weav_quick_ext* setup contains the following transforms:

mm9_ids calls the MM9 transform for current simulation.

copy_ids allows current to be copied from mm9_ids to id.m.

mm9_ib calls the MM9 transform for ib simulation.

copy_ib allows current to be copied from mm9_ids to ib.m.

weaval_extract calls the weak avalanche extraction functions.

quick_measure used by MM9_WEAVAL_EXT to initiate weak avalanche region measurements. Its functionality is the same as that of *quick_measure* in *lin_quick_ext*.

quick_ext/store contains miscellaneous data and transforms used during quick extraction. The *store* setup contains the following inputs and outputs:

index	An input definition used to set up array sizes
vdsids	An array containing the drain voltage offsets to be used by MM9_SAT_EXT for Ids measurements
vgsids	An array containing the gate voltages to be used by MM9_SAT_EXT for Ids measurements

vdsgds	An array containing the drain voltage offsets to be used by MM9_SAT_EXT for gds measurements
vgsgds	An array containing the gate voltages to be used by MM9_SAT_EXT for gds measurements

The *store* setup contains the following transforms:

ideal_parameters used to copy the present miniset parameters into the transform array.

restore_ideal_parameters used to set the miniset parameters to the values stored in the array *ideal_parameters*

working_parameters used to copy the present miniset parameters into the transform array

restore_working_parameters used to set the miniset parameters to the values stored in the array

print_par a call to MM9_SAVE_SPARS that appends the list of miniset variables to the file whose name is held in the model variable GEOMFILE.

quick_extraction_setup used to specify the quick extraction setup details including options and bias voltages. It can be used as an alternative to entering these details from the keyboard. You can make multiple copies of this transform (with different names) to store the setup information for frequently used processes. The setups in the present transform apply to a typical 5V process.

The new quick extraction functions control all aspects of quick extraction, that is, determining the bias levels to be applied to the device, initiating measurements, and performing calculations to extract the appropriate parameters.

MM9_LIN_EXT	Performs the linear region parameter extractions
MM9_STH_EXT	Performs the subthreshold parameter extractions
MM9_SAT_EXT	Performs the saturation parameter extractions including output conductance
MM9_WEAVAL_EXT	Performs the weak avalanche (substrate current) parameter extractions

The following arrays in *quick_ext/store* control the applied drain and gate biases:

vgsids	gate voltages for ids measurement
vdsids	drain voltages for ids measurement
vgsgds	gate voltages for gds measurement
vdsgds	drain voltages for gds measurement

Note that Vds is never allowed to have a value of less than 0.1V during saturation region quick extraction measurements.

The dutx DUT

The variable table of *dutx* contains the miniset parameters and the quantities VT1, VT2, and VT3, which are used to store the measured threshold voltages at the three back-biases used for the saturation and subthreshold measurements.

dutx / measure_vt performs a linear region measurement, that is, Ids vs. Vgs for a low value of Vds to determine the threshold voltage of the devices at the three values of Vbs used for subsequent measurements. An estimate of these threshold voltages is necessary to establish the gate biases for the saturation and subthreshold measurements.

The *measure_vt* setup contains the following transforms:

id_fit estimates Vt. It looks for the point of maximum transconductance, fits a straight line in the neighborhood of this point and estimates the threshold voltage from the intercept of this line with the Vgs axis. The output of this transform is the calculated current, based on the resulting transconductance and threshold voltage for display on the *vt_fit* plot.

calc_vt invokes *id_fit* for each of three Ids - Vgs curves measured in the setup. This transform also rounds the Vt values to the nearest 10mV.

mm9_ids calls the MM9 transform to evaluate the model current.

copy_sim_to_meas Copies the current generated by the MM9 transform into the measured array. It is used for making sample *measured* data. It uses the MM9_COPY C transform, which is necessary to enable data to be copied into a measured array. The variables table of *measure_vt* contains two quantities *VT_FIT* and *CURVE* where *CURVE* points to the curve that *calc_vt* is working on at a given time and *VT_FIT* is the threshold voltage associated with this curve.

dutx/idvg performs the measurements required for extraction of the linear region parameters. The *idvg* setup contains the following transforms:

The *idvg* setup contains the following transforms:

mm9_ids calls the MM9 transform to evaluate the model current.

set_dimensions Sets the correct values for W, L and MULT. For the measurement of any device, *dutx* is first copied to a new DUT. Then the dimension information in this DUT has to be set to correct values.

tid_lin converts the measured data to make a target array for the linear region extractions. It is common practice in Philips to filter any data points with current less than 10% of maximum when doing the linear region optimizations.

This transform mimics this procedure by setting any points less than 10% of maximum to a value of 0.5*IMIN. Because IMIN will be used to set an optimization floor, the resulting data points are ignored.

calc_all causes all the currents in the DUT to be re-evaluated with calls to MM9.

print_par calls the *MM9_SAVE_SPARS* transform that writes the miniset parameters to a file.

copy_sim_to_meas copies the current generated by the MM9 transform into the measured array. It is used for making sample *measured* data. It uses the *MM9_COPY* C transform which is necessary to enable data to be copied into a measured array.

set_par_from_quick_ext transfers miniset parameter values from the DUT *quick_ext* to a DUT containing the conventional optimization-type measured data. (found in *idvg* setup of *dutx).*

dutx/idvd1 performs the saturation region measurements for the first Vbs value (0V) that are needed for the optimization of the output conductance and saturation parameters.

The *idvd1* setup contains the following transforms:

gds is a call to the derivative function to evaluate the derivative of the measured current.

mm9_gds is a call to the derivative function to evaluate the derivative of the simulated current

mm9_ids calls the MM9 transform to evaluate the model current.

copy_sim_to_meas copies the current generated by the MM9 transform into the measured array. It is used for making sample *measured* data. It uses the *MM9_COPY* C transform which is necessary to enable data to be copied into a measured array.

set_vth stores the threshold voltage in the setup variable VTH.

dutx/idvd2, idvd3 perform the saturation region measurements at the two non-zero Vbs values. The data in these setups is not used during the parameter optimization sequences but is used as an extra check on model accuracy.

The *idvd2, idvd3* setups contain the following transforms:

gds is a call to the derivative function to evaluate the derivative of the measured current

mm9_gds is a call to the derivative function to evaluate the derivative of the simulated current

mm9_ids calls the MM9 transform to evaluate the model current.

copy_sim_to_meas copies the current generated by the MM9 transform into the measured array. It is used for making sample *measured* data. It uses the *MM9_COPY* C transform which is necessary to enable data to be copied into a measured array.

dutx/subvt1 performs the subthreshold measurements for the first value of Vbs (0V). These measurements are used for the subthreshold optimizations at Vbs = 0V.

The *subvt1* setup contains the following transforms:

mm9_ids calls the MM9 transform to evaluate the model current.

abs_vg is a call to the equation transform to calculate the absolute value of Vgs. This is necessary for the plot *logidvg_vbs* which shows the subthreshold current at non-zero Vbs values.

tid_svt generates target current values for subthreshold optimization. The main purpose is to eliminate data that could lie on the noise floor. It evaluates the *transconductance* on a log scale and eliminates points that have a *transconductance* of less than 70% of maximum on the low current side of the maximum point by setting their value to 0.5*IMIN.

copy_sim_to_meas copies the current generated by the MM9 transform into the measured array. It is used for making sample *measured* data. It uses the *MM9_COPY* C transform which is necessary to enable data to be copied into a measured array.

set_vth stores the threshold voltage in the setup variable VTH.

dutx/subvt2, **subvt3** enable measurement of subthreshold data for non-zero Vbs values that are required for the non-zero Vbs subthreshold optimizations.

The *subvt2, subvt3* setups contain the following transforms:

mm9_ids calls the MM9 transform to evaluate the model current.

abs_vg is a call to the equation transform to calculate the absolute value of Vgs. This is necessary for the plot *logidvg_vbs* which shows the subthreshold current at non-zero Vbs values.

tid_svt generates target current values for subthreshold optimization. The main purpose is to eliminate data that could lie on the noise floor. It evaluates the *transconductance* on a log scale and eliminates points that have a *transconductance* of less than 70% of maximum on the low current side of the maximum point by setting their value to 0.5*IMIN.

copy_sim_to_meas copies the current generated by the MM9 transform into the measured array. It is used for making sample *measured* data. It uses the *MM9_COPY* C transform which is necessary to enable data to be copied into a measured array.

set_vth stores the threshold voltage in the setup variable VTH.

dutx/ibvg allows the measurement of substrate (avalanche) current needed for the extraction of the substrate (avalanche) current parameters.

The *ibvg* setup contains the following transforms:

mm9_isub calls the MM9 transform to evaluate the avalanche current

copy_sim_to_meas copies the current generated by the MM9 transform into the measured array. It is used for making sample *measured* data. It uses the *MM9_COPY* C transform which is necessary to enable data to be copied into a measured array.

Macros

Macros control the overall extraction sequence.

SETUP Lets you provide setup information to describe the device type, dimensions, and matrix connections if appropriate, the bias voltages used, the nominal measurement temperature

and the measurement temperatures for temperature coefficient extraction. You can also specify minimum current and conductance levels for extraction. You can use setup to modify existing information as well as specify new information. The setup information is held in the model variable table of *MM9* and in the *devices* and *all_temp_ext* setups of the *extract* DUT. Any information that can be represented by a single value is held in the variable table, while information represented as an array is held in the setups. When SETUP is run, the information is first read from the existing IC-CAP arrays or variables. At the end of SETUP, the information is written back into the IC-CAP tables or arrays. SETUP also builds optimization tables for use in the *maxiset* and temperature extractions and puts them in the setups *scaled_ext*, *single_temp_ext*, and *all_temp_ext*.

measure Controls the measurement sequence for all specified devices. The macro prompts you to specify whether you want to measure the devices at the nominal temperature or at another temperature. The template for the measurements is located in *dutx*.

When you measure devices at the nominal temperature, *dutx* is copied as *dut1, dut2*, etc., for each device specified, and then the measurement transforms are invoked in each of these new DUTs.

When you measure devices at non-nominal temperatures, a new model is created for each specified temperature by copying *mm9_tempx* to a new model, *mm9_tx,* where x is a number representing the temperature.

extract_one_miniset Invokes the miniset extraction sequence for one device. It is a special case of the *extract_all_minisets* macro.

extract_all_minisets Controls the miniset extraction for all the devices measured at the nominal temperature. Miniset extraction consists of a series of optimizations that act on the miniset parameters. These miniset parameters are stored as DUT variables in the individual DUTs. The template for the extraction sequence is held in the setup *extract/single_ext*. As the miniset parameters for each DUT are being extracted, the setup *extract/single_ext* is first copied into the DUT. The

optimizations are then performed and the *single_ext* setup is then deleted from the DUT. This procedure was implemented to prevent multiple copies of what should be the same extraction sequence.

extract_maxiset Invokes the extraction of the maxiset parameters, i.e., the normal MOS Model 9 parameters at nominal temperature. First each of the miniset parameter sets is written to a file (whose name is given by the variable GEOMFILE) and then the transform *MM9_GEOMSCAL* is called. This reads the miniset parameters from the file just created and performs a least-squares fitting to obtain the maxiset parameters. This function writes the new parameter values into the parameter list and creates plots in the *par_vs_L*, *par_vs_W,* and *par_vs_R* setups of *extract* showing the variation of the miniset parameters with geometry and the fitting of this variation achieved by the maxiset parameters. Finally all the nominal devices are resimulated using the new maxiset parameters.

optimize_maxiset Calls the optimization sequence for the maxiset parameters at the nominal temperature. The extraction sequence itself is controlled by the transform *extract/scaled_ext/sca_opt*. After the optimization, all the devices are resimulated using the new model parameters.

display_parameter_vs_geometry_plots Displays plots of the chosen miniset parameters vs. geometry.

simulate_using_extended_equations Causes all the DUTs at the nominal temperature to be resimulated using the extended equations as would be used in a circuit simulator.

optimize_at_one_temperature Prompts you to specify the temperature of interest, calls the *extract/single_temp_ext/single_temp_opt* transform to perform optimizations of the temperature sensitive parameters at the chosen temperature, and then causes all the devices at this temperature to be resimulated using the new parameters. You would typically execute this macro once for each non-nominal temperature being used.

extract_temperature_coefficients Controls the extraction of the temperature coefficients that are valid over the full range of temperatures. First the temperature-sensitive parameters at all the temperatures are written to a file whose name is given by the variable *TEMPFILE*. Then the function MM9_TEMPSCAL is called which reads the parameters from the file just created and extracts the temperature coefficients using least-squares fitting. The DUTs at the non-nominal temperatures are then resimulated with the new parameters.

optimize_temperature_coefficients Calls the optimization sequences in *extract/all_temp_ext* to optimize the temperature coefficients for all the devices measured at the non-nominal temperatures. Each such device is resimulated with the new parameters when the optimizations are complete.

display_parameter_vs_temperature_plots Displays plots of specified parameters versus temperature.

quick_extraction_one_dut This asks the user to specify a DUT number (one of the devices already specified in setup) and then performs the quick extraction procedures on these. The measurements are performed in the *quick_ext* DUT and the miniset parameters extracted are placed in this DUT also. Therefore performing a quick extraction on a device will overwrite any data or miniset parameters in *quick_ext* associated with a previous device. Therefore, performing a quick extraction does not create any new data structures in IC-CAP. This choice to consider the quick extraction data as temporary and not to create new data structures for every device measured was made to keep the quick extraction time to a minimum and to avoid the possibility of generating an unmanageable model size when IC-CAP is being used to gather volume data (i.e. hundreds or more model sets) for statistical analysis.

test_quick_ext_with_ideal_pars This macro is used to test the quick extraction algorithms using synthetic data generated from a previously extracted/optimized set of miniset parameters.

make_extra_par_vs_geometry_plots This macro is used to create parameters versus length plots for a user-specified width and parameter versus width plots for a user-specified length. This is useful if the device set includes more than one “L-array” and more than one “W-array.”

display_extra_par_vs_geometry_plots Displays plots of the chosen miniset parameters versus L2 and W2.

read_data_from_directory Reads data previously stored in a subdirectory under the current working directory.

write_data_to_directory Writes data to a subdirectory under the current working directory.

Parameter Extraction

The purpose of parameter extraction is to determine the *maxiset* parameters needed to characterize a particular process. The implementation of MM9 in IC-CAP allows the extraction of all the model parameters that control DC behavior over a wide temperature range. The aim of this implementation is to extract values for parameters 1 through 70 in section 4.4, "List of scaling and reference parameters" of the Philips MOS Model 9 documentation (see Reference [4] at the end of this chapter).

The main extraction sequence is defined as a set of optimization transforms with a special function (MM9GEOMSCAL) used to determine a first-guess for the *maxiset* parameters by regression.

The main steps for parameter extraction are as follows:

1 Measure several devices at nominal temperature.

2 For each device, extract values for parameters 1 through 21 in section 4.5, “List of Parameters for an individual transistor” of the Philips MOS Model 9 documentation (see Reference [4] at the end of this chapter). These parameters are referred to as the *miniset* parameters. In practice, this step consists of a series of optimizations on the data for the individual devices.

   a Initialize parameter values

   Choose 1 or 2 body-effect factors

   Set ETAM, ETAGAM and ETADS

   b Linear $I_{ds} - V_{gs}$ data

   Optimize BET, THE1 and VTO for $V_{sb} = 0V$

   Optimize KO, (K, VSBX) and THE2 for all V_{sb}

   c Subthreshold $I_{ds} - V_{gs}$ data for $V_{sb} = 0V$

   Optimize GAMOO, MO and ZET1

   d Saturation $g_{ds} - V_{ds}$ data for $V_{sb} = 0V$

   Optimize VP for large device

Optimize GAM1 and ALP for other devices

e Saturation I_{ds} – V_{ds} data for V_{sb} = 0V

Optimize THE3

f Substrate current (I_{sub} – V_{gs}) data for V_{sb} = 0V

Optimize A1, A2 and A3

g Repeat steps b through f

h Subthreshold I_{ds} – V_{gs} data for all V_{sb}

Optimize VSBT

3 Apply the geometry scaling rules to the parameter sets generated in the previous step, and generate the full set of device parameters at the nominal temperature. (In practice, this step consists of a least-squares fitting procedure followed by optimizations on all the devices at nominal temperature.) This set of parameters is referred to as the *maxiset*.

- An initial estimate is obtained by fitting the scaling rules directly to the *miniset* parameters. This step also sets the parameters ETAALP, ETAGAMR, ETAMR, ETAZET, and ETADSR to their correct constant values.
- The resulting parameters are optimized to the measured characteristics of all the devices in the set.

4 For each temperature above or below the nominal, extract values of the temperature-sensitive parameters appropriate to this temperature. (In practice, this step consists of a series of optimizations on the devices measured at a particular non-nominal temperature.)

5 Apply the temperature scaling rules to the sets of parameters extracted in the previous step to generate the temperature coefficients of the model. (In practice, this step consists of a least-squares fitting followed by optimizations on all the devices measured at the non-nominal temperature.)

Data Organization

For extraction of MOS Model 9 parameters, I-V data is measured in accordance with the recommendations of the Philips' report NL-UR 003/94 “MOS Model 9.” This is basically:

- Linear region data:
  - I_{ds} – V_{gs} for low V_{ds}, range of V_{sb}
  - V_{gs} > V_{th}
- Subthreshold region data:
  - I_{ds} – V_{gs} for range of V_{ds} and V_{sb}
  - V_{gs} low to just above V_{th}
- Saturation characteristics:
  - I_{ds} – V_{ds} for range of V_{gs} and V_{sb}
  - includes one curve at V_{gs} = V_{th} + 100 mV
- Output conductance data:
  - g_{ds} – V_{ds} (derivative of I_{ds} – V_{ds} data)
- Substrate current data:
  - I_{sub} – V_{gs} for range of V_{ds} and V_{sb} = 0V

Scaling Rules

This section describes the scaling rules applied to individual parameters.

ALP

$$ALP = ALPR + \left(\frac{1}{L_{eff}^{ETAALP}} - \frac{1}{L_{er}^{ETAALP}}\right) SLALP + \left(\frac{1}{W_{eff}} - \frac{1}{W_{er}}\right) SWALP$$

where ETAALP = 0 or 1.

BET

$$\mathrm{BET} = \mathrm{BETSQ} \cdot \frac{W - dW}{L - dL}$$

where dW = *2WOT* – *WVAR* and dL = *2LAP* – *LVAR*

Here, BETSQ, dW, and dL can be extracted by a nonlinear fit to the *miniset* parameter BET versus W and L.

GAMOO, **MO**, **ZET1** and **VSBT**

$$P(W,L) = P_R + \left(\frac{1}{L_{eff}^n} - \frac{1}{L_{er}^n}\right) S_L$$

where n can have the value of 2, 0.5, ETAZET (0.5 or 1) or 1.

Here, the reference parameter P_R and the scaling coefficients S_L and S_W can be extracted by linear regression. The quantities W_{er} and L_{er} are the effective width and length of a reference device you choose.

KO, **K**, **VSBX**, **THE1**, **THE2**, **THE3**, **GAM1**, **A1**, **A2**, and **A3**

$$P(W,L) = P_R + \left(\frac{1}{L_{eff}} - \frac{1}{L_{er}}\right) S_L + \left(\frac{1}{W_{eff}} - \frac{1}{W_{er}}\right) S_W$$

VP

$$VP = VPR\left(\frac{L_{eff}}{L_{er}}\right)$$

VTO

$$VTO = VTOR + \left(\frac{1}{L_{eff}} - \frac{1}{L_{er}}\right) SLVTO + \left(\frac{1}{L_{efr}^2} - \frac{1}{L_{er}^2}\right) \mathrm{SL2VTO} + \left(\frac{1}{W_{eff}} - \frac{1}{W_{er}}\right) SWVTO$$

Device Geometries

The recommended criteria for selecting devices for extraction is illustrated in the following figure where *L-array* represents a set of devices with the same width but different lengths and *W-array* represents a set of devices with the same length but different widths.

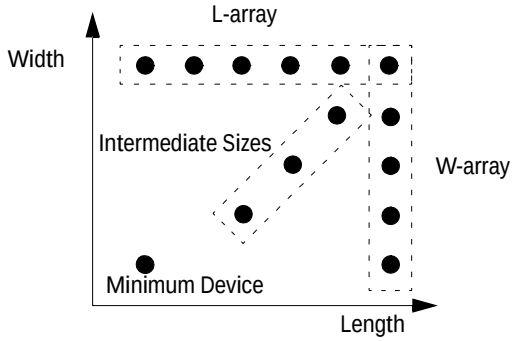


Figure 176 Device Size Selection

A quantity R_{eff} is defined to aid visualization of the scaling rules. For devices on the L and W arrays respectively:

$$\frac{1}{R_{eff}} = \frac{1}{L_{eff}} \quad \text{or} \quad \frac{1}{R_{eff}} = \frac{1}{W_{eff}}$$

For the other devices:

$$\frac{1}{R_{eff}} = \sqrt{\frac{1}{L_{eff}^2} + \frac{1}{W_{eff}^2}}$$

Optimizing

The steps below represent the basic maxiset optimization sequence.

1 Linear I_{ds} – V_{gs} data

V_{sb}=0V: Optimize VTOR, SLVTO, SL2VTO, SWVTO, BETSQ, THE1R, SLTHE1R, and SWTHE1.

Varying V_{sb}: Optimize KOR, SLKO, SWKO, (KR, SLK, SWK, VSBXR, SLVSBX, SWVSBX), THE2R, SLTHE2R, and SWTHE2.

2 Subthreshold I_{ds} – V_{gs} data

V_{sb} = 0V: Optimize GAMOOR, SLGAMOO, MOR, SLMO, ZET1R, and SLZET1.

3 Saturation g_{ds} – V_{ds} data

V_{sb} = 0V: Optimize GAM1R, SLGAM1, SWGAM1, ALPR, SLALP, SWALP, and VPR.

4 Saturation I_{ds} – V_{ds} data

V_{sb} = 0V: Optimize THE3R, SLTHE3R, and SWTHE3.

5 Substrate current (I_{sub} – V_{gs}) data

V_{sb} = 0V: Optimize A1R, SLA1, SWA1, A2R, SLA2, SWA2, A3R, SLA3, and SWA3.

6 Subthreshold I_{ds} – V_{gs} data

Varying V_{sb}: Optimize VSBTR and SLVSBT.

Optimization Transforms and Macros

The transforms described in this section are available with the DUT *extract*.

full_extract

The *full_extract* transform controls the optimization sequence for miniset extraction. It can be found under the setup *single_ext*.

1 Initialize parameters (*par_init*)

2 Linear region fitting at Vbs = 0 (*lin_opt1*)

3 Linear region fitting at all Vbs (*lin_opt2* for 2 k-factor model) or (*lin_opt3* for 1 k-factor model)

4 Subthreshold fitting at Vbs = 0 (*subvt_opt1*)

5 Gds fitting at Vbs = 0 (*normal_gds_opt1* for most devices) or (*large_gds_opt1* for the large device)

6 Ids fitting for Vbs = 0 (*ids_opt1*)

7 Avalanche fitting for Vbs = 0 (*isub_opt1*)

8 Repeat steps 2 through 7

9 Subthreshold fitting for all Vbs (*subvt_opt2*)

sca_opt

The *sca_opt* transform controls the optimization sequence for maxiset extraction. It can be found under the setup *scaled_ext*.

1 Linear region fitting at Vbs = 0 (*sca_lin_opt1*)

2 Linear region fitting for all Vbs (*sca_lin_opt2* for the 2 k-factor option) or (*sca_lin_opt3* for the 1 k-factor option)

3 Subthreshold fitting at Vbs = 0 (*sca_subvt_opt1*)

4 Gds fitting at Vbs = 0 (*sca_gds_opt1*)

5 Ids fitting at Vbs = 0 (*sca_ids_opt1*)

6 Avalanche current fitting at Vbs = 0 (*sca_isub_opt1*)

7 Subthreshold fitting for all Vbs (*sca_subvt_opt2*)

single_temp_opt

The *single_temp_opt* transform controls the optimization sequence for the temperature-dependent parameters at a single non-nominal temperature. It can be found under the setup *single_temp_extract*.

1 Initialize variables (*temp_par_init*)

2 Linear fitting at Vbs = 0 (*single_temp_lin_opt1*)

3 Linear fitting at all Vbs (*single_tep_lin_opt2*)

4 Subthreshold fitting at Vbs = 0 (*single_temp_subvt_opt1*)

5 Ids fitting at Vbs = 0 (*single_temp_ids_opt1*)

6 Avalanche current fitting at Vbs = 0 (*single_temp_isub_opt1*)

optimize_temperature_coefficients

The *optimize_temperature_coefficients* macro controls the optimization sequence for the temperature coefficients by calling the transforms listed below (found under the *all_temp_extract* setup) in the order shown.

1 *all_temp_lin_opt1*

2 *all_temp_lin_opt2*

3 *all_temp_subvt_opt1*

4 *all_temp_ids_opt1*

5 *all_temp_isub_opt1*

The JUNCAP Model

The JUNCAP model represents the C-V and I-V behavior of the parasitic source and drain regions of MOSFET devices [5].

The JUNCAP model file contains four DUTs: *area*, *locos*, *gate* and *analysis*. The *area*, *locos*, and *gate* DUTs hold the data for the *area*, *locos* and *gate* test structures, respectively. The *analysis* DUT, and its associated setups, contains the transforms that control the parameter extraction strategies.

The area, locos, and gate DUTs

The *area*, *locos*, and *gate* DUTs all have the same structure, as shown in the following table.

Table 57 Parameters for *area*, *locos* and *gate* Test Structures

area	locos	gate
AB is set to AB1 LS is set to LS1 and LG is set to 0	AB is set to AB2 LS is set to LS2 and LG is set to 0	AB is set to AB3, LS is set to LS3 and LG is set to LG3

Each of these DUTs contain three setups: *cv*, *fwd_iv* and *rev_iv*.

cv This setup contains measured and simulated C-V data. It consists of the following:

va This input defines the voltage sweep for C-V measurement. It uses the variables CVSTART, CVSTOP and CVSTEP as defined in the *setup_details* macro.

cap This output holds the measured capacitance.

cap_sim This transform calls JUNCAP to evaluate the *simulated* capacitance.

make_cv_data This transform is used for making synthetic data for demonstration purposes. It performs a model evaluation using the existing parameter set by calling

cap_sim and then copies the resulting simulated data into the *m* part of the cap output. This macro assumes that the MOS Model 9 function MM9_COPY is available.

connect_cv Modify this transform to enable automatic connection to the *area* DUT for C-V measurements.

cv_plot This is a plot definition showing measured and simulated C-V data.

fwd_iv This setup contains the measured and simulated forward I-V data. It consists of the following:

va, vk These inputs define the anode and cathode voltages for forward I-V measurements. The variables FIVSTART, FIVSTOP and FIVSTEP control the voltage sweeps.

id The output current

id_sim A call to JUNCAP to evaluate the simulated current

make_iv_data A transform to make synthetic forward I-V data. The function MM9_COPY is used in this transform.

connect_fiv Modify this transform to enable automatic connection to the DUT for forward I-V measurements.

fwd_ivplot The plot definition for the forward I-V data

rev_iv This setup contains the measured and simulated reverse I-V data. It consists of the following:

va, vk These inputs define the anode and cathode voltages for reverse I-V measurements. The variables RIVSTART, RIVSTOP and RIVSTEP control the voltage sweeps.

id The output current

id_sim A call to JUNCAP to evaluate the simulated current

set_temp This transform sets the setup level variable TEMP to the model level variable TREVERSE. The reverse data may be measured at a different temperature, TREVERSE, than the forward I-V or C-V data. However, the JUNCAP function looks for a variable TEMP to determine the device temperature. Therefore TEMP is defined as a setup level variable in the

area/rev_iv, *locos/rev_iv*, *gate/rev_iv* and *analysis/rev_iv* setups. Thus, during simulations in these setups, the setup variable TEMP will supersede the model level variable TEMP.

make_iv_data A transform to make synthetic reverse I-V data. The function MM9_COPY is used is this transform.

connect_riv You can modify this transform to enable automatic connection to the DUT for reverse I-V measurements.

rev_ivplot The plot definition for the reverse I-V data.

The analysis DUT

In the *analysis* DUT, the dimensions AB, LS and LG are set to unity. The setups are *cv*, *fwd_iv* and *rev_iv*.

cv This setup controls the extraction of the C-V parameters and contains the following:

va This input definition for the anode voltage is the same as that in the *area/cv*, *locos/cv* and *gate/cv setups*.

cjbvn A transform that extracts (and holds) the normalized area sub-region contribution to capacitance from the measurements in the *area/cv*, *locos/cv* and *gate/cv* setups.

set_cjbr A transform that makes an initial approximation to the parameter CJBR by setting it to the value of cjbvn at the point where the anode voltage is closest to zero.

cjbvn_sim A transform that calls JUNCAP to evaluate the area sub-region component of capacitance.

fit_cjbvn An optimization definition that causes the parameters CJBR, PB and VDBR to be optimized with respect to the normalized area sub-region capacitance. The parameter limits are controlled by the following model variables, which you can change in the model variables table:

CJBR_MIN	CJBR_MAX	PB_MIN
PB_MAX	VDBR_MIN	VDBR_MAX

The data limits are controlled by the following variables, which are also in the model variables table:

CV_VMIN	CV_VMAX

cjsvn A transform that extracts (and holds) the normalized locos sub-region contribution to capacitance from the measurements in the *area/cv*, *locos/cv* and *gate/cv setups*.

set_cjsr A transform that makes an initial approximation to the parameter CJSR by setting it to the value of cjsvn at the point where the anode voltage is closest to zero.

cjsvn_sim A transform that calls JUNCAP to evaluate the locos sub-region component of capacitance

fit_cjsvn An optimization definition that causes the parameters CJSR, PS and VDSR to be optimized with respect to the normalized locos sub-region capacitance. The parameter limits are controlled by the following model variables, which you can change in the model variables table.

CJSR_MIN	CJSR_MAX	PS_MIN
PS_MAX	VDSR_MIN	VDSR_MAX

The data limits are controlled by the variables, which are also in the model variables table.

CV_VMIN	CV_VMAX

cjgvn A transform that extracts (and holds) the normalized gate sub-region contribution to capacitance from the measurements in the *area/cv*, *locos/cv* and *gate/cv* setups.

set_cjgr A transform that makes an initial approximation to the parameter CJGR by setting it to the value of cjgvn at the point where the anode voltage is closest to zero.

cjgvn_sim A transform that calls JUNCAP to evaluate the gate sub-region component of capacitance

fit_cjgvn An optimization definition that causes the parameters CJGR, PG and VDGR to be optimized with respect to the normalized gate sub-region capacitance. The parameter limits are controlled by the following model variables, which you can change in the model variables table.

CJGR_MIN	CJGR_MAX	PG_MIN
PG_MAX	VDGR_MIN	VDGR_MAX

The data limits are controlled by the following variables, which are also in the model variables table:

CV_VMIN	CV_VMAX

init_cv_pars A transform to initialize some of the C-V parameters before optimization begins. The parameters initialized are:

VDBR = VDSR = VDGR = 0.75

PB = PS = PG = 0.4

opt_all_cv An optimization definition that causes all the C-V parameters to be optimized with respect to the measured data in the *area/cv*, *locos/cv* and *gate/cv* setups. The parameters optimized are:

CJBR	CJSR	CJGR
VDBR	VDSR	VDGR
PB	PS	PG

The parameter limits are controlled by the following model variables:

CJBR_MIN	CJBR_MAX	CJSR_MIN	CJSR_MAX	CJGR_MIN
CJGR_MAX	VDBR_MIN	VDBR_MAX	VDSR_MIN	VDSR_MAX
VDGR_MIN	VDGR_MAX	PB_MIN	PB_MAX	PS_MIN

PS_MAX PG_MIN PG_MAX

The data limits are controlled by the following variables:

CV_VMIN CV_VMAX

set_unit_dimensions A transform that sets the dimensions AB, LS and LG in the *analysis* DUT to unity.

update_cv_curves A transform that resimulates all the C-V curves in the *area/cv*, *locos/cv*, *gate/cv* and *analysis/cv* setups.

cjb A plot definition for the normalized area sub-region contribution to capacitance.

cjs A plot definition for the normalized locos sub-region contribution to capacitance.

cjs A plot definition for the normalized gate sub-region contribution to capacitance.

fwd_iv This setup controls the extraction of the I-V parameters with respect to the forward I-V data. In the forward region at moderate and high applied voltages the diffusion current components dominate. However for low-applied voltages the generation components are also important. Therefore all the optimizations to the forward I-V curves target both the diffusion and generation parameters.

va, **vk** These input definitions for the anode and cathode voltages are the same as those in the *area/fwd_iv*, *locos/fwd_iv* and *gate/fwd_iv* setups.

ibn A transform that extracts (and holds) the normalized area sub-region contribution to forward current from the measurements in the *area/fwd_iv*, *locos/fwd_iv* and *gate/fwd_iv* setups.

ibn_sim A transform that calls JUNCAP to evaluate the area sub-region component of current.

fit_ibn An optimization definition that causes the parameters JSDBR, NB and JSGBR to be optimized with respect to the normalized area sub-region forward current. The parameter limits are controlled by the model variables, which you can change in the model variables table.

JSDBR_MIN	JSDBR_MAX	NB_MIN
NB_MAX	JSGBR_MIN	JSGBR_MAX

The data limits are controlled by the following variables, which are also in the model variables table.

FIV_VMIN	FIV_VMAX

isn A transform that extracts (and holds) the normalized locos sub-region contribution to forward current from the measurements in the *area/fwd_iv*, *locos/fwd_iv* and *gate/fwd_iv* setups.

isn_sim A transform that calls JUNCAP to evaluate the locos sub-region component of current.

fit_isn An optimization definition that causes the parameters JSDSR, NS and JSGSR to be optimized with respect to the normalized locos sub-region forward current. The parameter limits are controlled by the following model variables, which you can change in the model variables table.

JSDSR_MIN	JSDSR_MAX	NS_MIN
NS_MAX	JSGSR_MIN	JSGSR_MAX

The data limits are controlled by the following variables, which are also in the model variables table:

FIV_VMIN	FIV_VMAX

ign A transform that extracts (and holds) the normalized gate sub-region contribution to forward current from the measurements in the *area/fwd_iv*, *locos/fwd_iv* and *gate/fwd_iv* setups.

ign_sim A transform that calls JUNCAP to evaluate the gate sub-region component of current.

fit_ign An optimization definition that causes the parameters JSDGR, NG and JSGGR to be optimized with respect to the normalized gate sub-region forward current. The parameter limits are controlled by the following model variables, which you can change in the model variables table:

JSDGR_MIN	JSDGR_MAX	NG_MIN
NG_MAX	JSGGR_MIN	JSGGR_MAX

The data limits are controlled by the following variables, which are also in the model variables table.

FIV_VMIN	FIV_VMAX

init_iv_pars A transform to initialize some of the I-V parameters before optimization begins. The parameters initialized are:

JSDBR = 10n	JSDSR = JSDGR = 10f	NB = NS = NG = 1
JSGBR = 1u	JSGSR = JSGGR = 100p	

opt_all_fwd_iv An optimization definition that causes all the I-V parameters to be optimized with respect to the measured data in the *area/fwd_iv*, *locos/fwd_iv* and *gate/fwd_iv* setups. The parameters optimized are:

JSDBR	JSDSR	JSDGR	NB	NS
NG	JSGBR	JSGSR	JSGGR	

The parameter limits are controlled by the following model variables:

JSDBR_MIN	JSDBR_MAX	JSDSR_MIN	JSDSR_MAX
JSDGR_MIN	JSDGR_MAX	NB_MIN	NB_MAX
NS_MIN	NS_MAX	NG_MIN	NG_MAX
JSGBR_MIN	JSGBR_MAX	JSGSR_MIN	JSGSR_MAX,
JSGGR_MIN	JSGGR_MAX		

The data limits are controlled by the variables:

FIV_VMIN	FIV_VMAX

update_iv_curves A transform that resimulates all the I-V curves in the following setups:

area	fwd_iv, rev_iv
locos	fwd_iv, rev_iv
gate	fwd_iv, rev_iv
analysis	fwd_iv, rev_iv

ib A plot definition for the normalized area sub-region contribution to current

is A plot definition for the normalized locos sub-region contribution to current

ig A plot definition for the normalized gate sub-region contribution to current

rev_iv This setup controls the extraction of the I-V parameters with respect to the reverse I-V data. Current in the reverse region is dominated by the generation effects and so only the generation parameters are considered during these extractions.

va, **vk** These input definitions for the anode and cathode voltages are the same as those in the *area/rev_iv*, *locos/rev_iv* and *gate/rev_iv* setups.

ibn A transform that extracts (and holds) the normalized area sub-region contribution to reverse current from the measurements in the *area/rev_iv*, *locos/rev_iv* and *gate/rev_iv* setups.

ibn_sim A transform that calls JUNCAP to evaluate the area sub-region component of current.

fit_ibn An optimization definition that causes the parameter JSGBR to be optimized with respect to the normalized area sub-region reverse current. The parameter limits are controlled by the following model variables, which you can change in the model variables table.

JSGBR_MIN	JSGBR_MAX

The data limits are controlled by the following variables, which are also in the model variables table.

RIV_VMIN	RIV_VMAX

isn A transform that extracts (and holds) the normalized locos sub-region contribution to reverse current from the measurements in the *area/rev_iv*, *locos/rev_iv* and *gate/rev_iv* setups.

isn_sim A transform that calls JUNCAP to evaluate the locos sub-region component of current.

fit_isn An optimization definition that causes the parameter JSGSR to be optimized with respect to the normalized locos sub-region reverse current. The parameter limits are controlled by the following model variables, which you can change in the model variables table.

JSGSR_MIN	JSGSR_MAX

The data limits are controlled by the following variables, which are also in the model variables table.

RIV_VMIN RIV_VMAX

ign A transform that extracts (and holds) the normalized gate sub-region contribution to reverse current from the measurements in the *area/rev_iv*, *locos/rev_iv* and *gate/rev_iv* setups.

ign_sim A transform that calls JUNCAP to evaluate the gate sub-region component of current.

fit_ign An optimization definition that causes the parameter JSGGR to be optimized with respect to the normalized gate sub-region reverse current. The parameter limits are controlled by the following model variables, which you can change in the model variables table.

JSGGR_MIN JSGGR_MAX

The data limits are controlled by the following variables, which are also in the model variables table.

RIV_VMIN RIV_VMAX

set_temp This transform sets the setup level variable TEMP to the model level variable TREVERSE.

opt_all_rev_iv An optimization definition that causes all the generation parameters to be optimized with respect to the measured data in the *area/rev_iv*, *locos/rev_iv* and *gate/rev_iv* setups. The parameters optimized are:

JSGBR JSGSR JSGGR

The parameter limits are controlled by the following model variables:

JSGBR_MIN	JSGBR_MAX	JSGSR_MIN
JSGSR_MAX	JSGGR_MIN	JSGGR_MAX

The data limits are controlled by the following variables:

FIV_VMIN	FIV_VMAX

ib A plot definition for the normalized area sub-region contribution to current.

is A plot definition for the normalized locos sub-region contribution to current.

ig A plot definition for the normalized gate sub-region contribution to current.

Macros

This section describes the macros provided with the JUNCAP model.

setup_details This macro prompts you for various setup details. These details are stored in the model variables table. The current values of the variables are used as prompts so you can easily change one setting by executing the macro a second time and choosing OK to all questions except the change required. The information requested by this macro is as follows:

TR	Model parameter representing reference temperature
VR	Model parameter representing the reference voltage for parameter scaling (usually 0)
TEMP	Ambient temperature at which the forward I-V and the C-V curves will be measured
TREVERSE	The temperature at which the reverse I-V curves will be measured. If possible, this should be higher than TEMP to accentuate the current component from generation effects
AB1	The area of the area DUT
LS1	The locos perimeter length of the area DUT

AB2	The area of the locos DUT
LS2	The locos perimeter length of the locos DUT
AB3	The area of the gate DUT
LS3	The locos perimeter length of the gate DUT
LG3	The gate perimeter length of the gate DUT
CONNECT_CV	This variable is set depending on whether the user wishes to use manual or automatic connections for C-V measurement.
CONNECT_FIV	This variable is set depending on whether the user wishes to use manual or automatic connections for forward I-V measurement.
CONNECT_RIV	This variable is set depending on whether the user wishes to use manual or automatic connections for reverse I-V measurement.
CVSTART	The start voltage for C-V sweeps
CVSTOP	The stop voltage for C-V sweeps
CVSTEP	The voltage step for C-V sweeps
FIVSTART	The start voltage for forward I-V sweeps
FIVSTOP	The stop voltage for forward I-V sweeps
FIVSTEP	The voltage step for forward I-V sweeps
RIVSTART	The start voltage for reverse I-V sweeps
RIVSTOP	The stop voltage for reverse I-V sweeps
RIVSTEP	The voltage step for reverse I-V sweeps
DATASOURCE	If set to M, measurements will be taken. Otherwise data will be generated from simulations
DISPLAYPLOTS	If set to Y, plots are displayed during measurement and optimizations. Otherwise they are not (unless they have previously been displayed and not closed).

The *setup_details* macro also calls the transform *set_unit_dimensions*, which sets the dimensions in the *analysis* DUT to unity.

measure_cv This macro causes a measurement to be taken in the *cv* setups of the *area*, *locos* and *gate* DUTs. At the end of these measurements, you are prompted to specify the following information:

CV_VMI	The lower voltage limit for optimizations with respect to the C-V data
CV_VMAX	The upper voltage limit for optimizations with respect to the C-V data

measure_forward_iv This macro causes a measurement to be taken in the *fwd_iv* setups of the *area*, *locos* and *gate* DUTs. At the end of these measurements. you are prompted to specify the following information:

FIV_VMIN	The lower voltage limit for optimizations with respect to the forward I-V data
FIV_VMAX	The upper voltage limit for optimizations with respect to the forward I-V data

measure_reverse_iv This macro causes a measurement to be taken in the *rev_iv* setups of the *area*, *locos* and *gate* DUTs. At the end of these measurements, you are prompted to specify the following information:

RIV_VMIN	The lower voltage limit for optimizations with respect to the reverse I-V data
RIV_VMAX	The upper voltage limit for optimizations with respect to the reverse I-V data

extract_cv_pars This macro controls the extraction of the C-V parameters by splitting the C-V data into the *area*, *locos* and *gate* contributions and optimizing the parameters to these. At the end of the extractions, the simulated C-V arrays in the *area*, *locos*, *gate* and *analysis* DUTS are updated with the new parameters.

extract_fwd_iv_pars This macro controls the extraction of the forward I-V parameters by splitting the forward I-V data into the *area*, *locos* and *gate* contributions and optimizing the parameters to these. At the end of the extractions, the simulated I-V arrays (forward and reverse) in the *area*, *locos*, *gate* and *analysis* DUTS are updated with the new parameters.

extract_rev_iv_pars This macro controls the extraction of the reverse I-V parameters by splitting the reverse I-V data into the *area*, *locos* and *gate* contributions and optimizing the parameters to these. At the end of the extractions, the simulated I-V arrays (forward and reverse) in the *area*, *locos*, *gate* and *analysis* DUTS are updated with the new parameters.

opt_all_cv This macro controls the optimization of the full set of C-V parameters with respect to the measured data in the *area*, *locos* and *gate* DUTs. At the end of the extractions, the simulated C-V arrays in the *area*, *locos*, *gate* and *analysis* DUTS are updated with the new parameters.

opt_all_fwd_iv This macro controls the optimization of the full set of I-V parameters with respect to the measured forward I-V data in the *area*, *locos* and *gate* DUTs. At the end of the extractions, the simulated I-V arrays (forward and reverse) in the *area*, *locos*, *gate* and *analysis* DUTS are updated with the new parameters.

opt_all_rev_iv This macro controls the optimization of the generation current parameters with respect to the measured reverse I-V data in the *area*, *locos* and *gate* DUTs. At the end of the extractions, the simulated I-V arrays (forward and reverse) in the *area*, *locos*, *gate* and *analysis* DUTS are updated with the new parameters.

simulate_all_curves This macro allows all the curves to be resimulated.

set_new_TR This macro allows the model parameters to be recalculated for a new reference temperature.

read_data_from_directory Reads data previously stored in a subdirectory under the current working directory.

write_data_to_directory Writes the data to a subdirectory under the current working directory.

General Extraction Methodology

The JUNCAP model extraction methodology assumes that the parasitic source and drain regions consist of three sub-regions:

- The area of the source/drain

  - On an IC layout, this is the area of the source/drain active region. This area is labeled AB and has dimensions of m^2.
- The LOCOS edge
  - On an IC layout, this is the perimeter of the source/drain region that is shared with the LOCOS edge. This perimeter is labeled LS and has dimensions of m.
- The gate edge
  - On an IC layout, this is the perimeter of the source/drain region that is shared with the gate polysilicon edge. This perimeter is labeled LG and has dimensions of m.

Parameters are specified for each of these three sub-regions separately. To enable these parameters to be uniquely determined, at least three different source/drain regions must be measured with various dimensions for the three sub-regions. The present implementation assumes that three different test structures labeled *area*, *locos* and *gate* will be measured, as shown in the following table.

Table 58 Test Structures for JUNCAP Model

DUT	AB	LS	LG
area	AB1 large	LS1 small	LG1 zero
locos	AB2 small	LS2 large	LG2 zero
gate	AB3 intermediate	LS2 intermediate	LG3 non-zero

The parameters AB1, AB2 and AB3 are the areas associated with the three DUTs; LS1, LS2 and LS3 are the LOCOS perimeters; and LG1, LG2, and LG3 are the gate perimeters. In the area DUT, the contribution of the area sub-region is assumed to be large while the contributions of the locos and gate sub-regions are small or zero. The locos DUT is assumed to have a larger contribution from the locos sub-region and the gate DUT has a non-zero contribution from the gate sub-region.

The capacitance associated with any DUT is considered to be the sum of the contributions of the three sub-regions. For example, for the DUT gate, the capacitance at any voltage V is given by:

C(V) = C_AREA(V).AB3 + C_LOCOS(V).LS3 + C_GATE(V).LG3

where C_AREA(V),C_LOCOS(V) and C_GATE(V) are the normalized contributions of the area, locos and gate sub-regions at voltage V.

With respect to current, JUNCAP includes two mechanisms: diffusion and generation. These are described separately for each sub-region so that for the DUT gate, the current flow at any voltage V is given by:

I(V) = (ID_AREA(V) + IG_AREA(V)).AB3

+ (ID_LOCOS(V) + IG_LOCOS(V)).AL3

+ (ID_GATE(V) + IG_GATE(V)).AG3

where ID_AREA(V) and IR_AREA(V) are the normalized contributions of the diffusion current and generation current, respectively, for the area sub-region at voltage V with similar notation being used for the locos and gate sub-regions.

Once the three DUTs have been measured, a set of simultaneous equations can be solved that allows the contributions of the area, locos and gate sub-regions to be separated and normalized. Parameter extraction then proceeds by optimizing the relevant parameters to each of the sub-region contributions in turn. Finally, the model parameters may be fine-tuned by optimization with respect to the directly-measured data in the area, locos and gate DUTs.

For the case of a well diode, you should specify that there is no *gate* test structure by setting the variable AB3 to zero. With AB3=0, the *gate* device will be ignored during measurements and optimizations.

References

1 “Compact MOS modeling for analog circuit simulation” (IEDM '93)

2 “The high-frequency analogue performance of MOSFETs” (IEDM '94)

3 “Circuit Sensitivity Analysis in Terms of Process Parameters” (SISDEP '95)

4 Unclassified report, NL-UR 003/94, R.M.D.A. Velghe, D.B.M. Klaassen, F. M. Klaassen, Philips Research Laboratories, June, 1995

5 Unclassified report, NL-UR 028/95, R.M.D.A. Velghe

9 Circuit Modeling

Definition of an IC-CAP Circuit 538
IC-CAP Circuit Modeling Operations 539
Defining a Circuit 540
Circuit Measurement 543
Circuit Parameter Extraction 545
Circuit Simulation 549
Test Circuits 551
Hierarchical Modeling 553
Functional Circuit Blocks 555
References 566

Circuit modeling is a natural extension of single device modeling. With IC-CAP's flexible structure, it is as easy to measure and characterize a multicomponent circuit as a single device. This chapter provides details for performing circuit modeling; typical applications are also provided to use as a guide for meeting specific circuit modeling requirements.

Definition of an IC-CAP Circuit

IC-CAP defines a circuit as any connection of two or more components. Previous chapters have dealt primarily with single devices such as bipolar, GaAs or MOS transistors. An IC-CAP circuit can be a simple two-resistor voltage divider or a complex operational amplifier or A/D converter.

The circuit, like a single device, is specified in the Circuit folder of the model window using SPICE compatible circuit definition syntax. All circuit decks in IC-CAP begin with the *.SUBCKT* subcircuit definition and end with the *.ENDS* statement. Circuit modeling allows more accurate solutions to many single device modeling requirements and expands the level of systems modeling possible.

IC-CAP Circuit Modeling Operations

With IC-CAP, every type of characterization operation performed on a single component can also be performed on a circuit. IC-CAP allows easy measurement of circuit characteristics, extraction and optimization of model parameters, and simulation of the circuit's performance. Measurement and simulation operations use the same setup information as single components. Extraction and optimization operations enable more options for methods of obtaining model parameters. These operations can be performed on the circuit as a whole or on any sub-component of the circuit. This is explained in the section, "Circuit Parameter Extraction" on page 545.

Defining a Circuit

The process of defining a circuit in IC-CAP is similar to defining a single device. The main difference is the interconnection of the components and the use of subcircuit lines to define the circuit block. For detailed information on defining circuits, refer to the appropriate *Reference* chapter (Chapter 3, "SPICE Simulators," Chapter 4, "SPECTRE Simulator," Chapter 5, "Saber Simulator," Chapter 6, "MNS Simulator").

Supported Circuit Components

Circuits in IC-CAP support the standard components that can be simulated with SPICE:

Passive elements R, L, C, Transmission lines

Semiconductors Bipolar, MOS, GaAs, JFET, Diode

Sources V, I, VCVS, VCIS, ICVS, ICIS

The syntax for defining a circuit in IC-CAP is similar to a SPICE simulation input deck. Each line contains a component, its node numbers, value, and (if applicable) an associated model name reference. Proper specification and use of these components is critical to the success of circuit simulation and parameter extraction.

In general, independent voltage sources are specified as inputs within a given setup. This allows you to specify their values and use them in additional numerical or graphic analysis. Some of the differences between SPICE and IC-CAP circuit definitions are listed.

- The .OPTIONS statement (if used) must be the first line in the circuit description. All options must be on one line (no continuation).
- The next line of the circuit is .SUBCKT.
- A TITLE specification is automatically generated by IC-CAP and should not be included in the circuit definition.
- The last line of the circuit is .ENDS

- An .END statement is automatically generated by IC- CAP and should not be included in the circuit definition.

The following figure shows an example circuit description. This circuit defines the input section of an ECL OR/NOR logic gate. (Figure 178 shows the schematic.) This circuit is referenced several times in this chapter. You can create it using the circuit editor or read it from the file *$ICCAP_ROOT/data/ECLornor.mdl*.

```
.SUBCKT ECLORNOR  1=IN1 2=IN2 3=OR 4=NOR
+ 5=VCC 6=VEE 7=VREF
* ECL OR/NOR LOGIC GATE
Q1   4  1  8  NPN1
Q2   4  2  8  NPN1
Q0   3  7  8  NPN2
.MODEL NPN1 NPN
+ IS = 2E-14  NF = 0.998  BF = 120
+ RB = 225  CJC = 300p  TF = 20p
.MODEL NPN2 NPN
+ IS = 4E-14  NF = 0.998  BF = 120
+ RB = 110  CJC = 530p  TF = 18p
RL1  5  4  300
RL0  5  3  300
RIEE 8  6  1.2K
.ENDS
```

Figure 177 Circuit Description for an ECL OR/NOR Logic Gate

When you enter the circuit description in the Circuit folder of the model window, moving the mouse out of the Circuit folder automatically causes the circuit to be parsed, that is, the specified circuit elements are read and entries are created for them in Model Parameters. When they are added initially, they assume the value specified in the circuit description. To change a value subsequently, you must change it in Model Parameters. To change all entries in Model Parameters to the values in the circuit description, choose **Reset**.

Note the difference in the Parameters table parameter names for a transistor in a circuit. In a single transistor circuit, the model parameter names of the transistor are the entries in the Parameters table. In a multi- component circuit the transistor's model parameters must be associated with a specific model, so the parameters take on a prefix of that model's name. Thus, the forward Beta model parameter *BF* for a model named *NPN1* is

listed in the Parameter Editor as *NPN1.BF*. In the example above, transistors *Q1* and *Q2* both use the *NPN1* model, while transistor *Q0* uses the *NPN2* model.

Circuit Measurement

The process of measuring a circuit in IC-CAP is identical to measuring a single device. The circuit stimuli and responses are specified in the input and output tables, respectively, of the Measure/Simulate folder. You can perform a measurement by clicking Measure in the Measure/Simulate folder the DUT or Setup levels. In performing measurements on circuits, there are several additional items not found in single component measurement.

Multiple Instrument Names

In measuring a single component, it is common to use only one DC source and measurement instrument because only four terminals are involved. The typical circuit can have more than four terminals and require several DC source and measurement instruments. Any number of instruments of the same or similar type can be connected to the circuit under test as long as they are entered in Hardware Setup. When using multiple instruments, each of their units must have a unique name.

Isolating Circuit Elements for Measurement and Extraction

The characterization of a circuit may require the measurement and modeling of several sub-circuit elements. The accuracy of the sub-circuit model generated is dependent upon how well that circuitry can be isolated from the rest of the overall circuit.

Examine the simplified schematic of the input to an ECL OR/NOR gate in Figure 178. The input stage of this circuit is a differential amplifier with a collector resistor in each leg and a resistor for a current source. It is possible to characterize the individual transistors in this circuit by selectively biasing only the terminals that make it active and that keep other parts of the circuit in an off or latent state. In this case, biasing IN1, VCC, and VEE turns on the circuit that contains RL1, Q1, and RIEE. These components have now been isolated so that their model parameters can be

extracted. This type of selective measurement allows characterization of individual or small groups of sub-circuit elements.

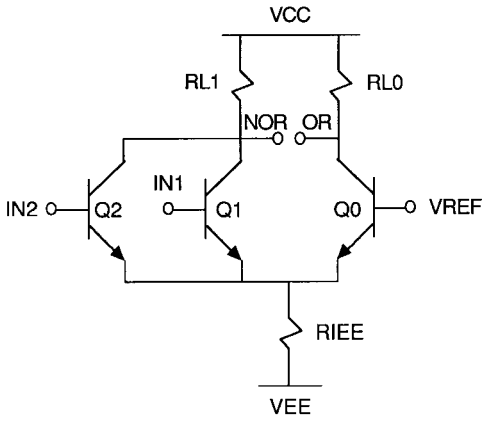


Figure 178 ECL OR/NOR Schematic Diagram

Circuit Parameter Extraction

Circuit parameter extraction is identical to single component parameter extraction through the use of Transforms. However, because circuits are custom in nature, most of the extraction routines must also be custom designed. With the availability of the Program function and optimize transforms, this is simple and quick to evaluate and execute. The critical factor in a successful circuit level parameter extraction is the ability to make a measurement and subsequent extraction involving only the dominant component parameters.

For a full model extraction of a single component, you will attain more accuracy if that device is available without any additional components connected to it. For most functional block level circuits however, a subset of the transistor model parameters is usually sufficient for studying circuit behavior.

Extracting Transistor Parameters Using Library Functions

In the explanation of a selective measurement on a sub-portion of a circuit in the previous section, Q1 and its neighboring resistors were isolated in the ECL logic gate. The forward active model parameters can be extracted from this measurement using the model extraction functions in the function list or by setting up a custom optimization. To access the functions, add transforms that use them to a setup that contains the measurement. It is possible to use the provided transistor extraction functions to obtain model parameters for devices connected into a larger circuit.

Because all models in a circuit have model parameters in the Parameters table with the model's name as a prefix, IC-CAP must be told which model to use with the extraction transforms. This is easily done by setting a variable in the model level variable table. Enter a variable in the table called EXTR_MODEL and set its value to the name of the transistor whose parameters are being extracted. When the extraction transforms are executed, IC-CAP refers to the correct Parameters table entry as it writes the extracted value back to

the table. For example, to use a function list transform on the model *NPN1* mentioned above, add the following to the model level variable table:

```
EXTR_MODEL      NPN1
```

Each time another transistor is used for an extraction, place its name in the value field. A more efficient method of extracting individual transistor models is to create an individual setup for each device. The variable table at the setup level can then include the EXTR_MODEL entry, keeping the transistor extraction local to that setup. This can also be done in an analogous way at the DUT level.

It is sometimes necessary to specify the particular DUT in a circuit that should be used in an extraction routine. For example, in a circuit that contains two MOSFETs there are two different sets of geometry parameters (L and W). For the extraction to work correctly, the EXTR_DUT variable must be set to the name of the transistor with the correct geometry parameters. Therefore, to characterize transistor *M1*, which uses model *NMOS1*, add the following to the model level variable table:

```
EXTR_DUT        M1
EXTR_MODEL      NMOS1
```

Situations where EXTR_DUT must be set can also arise when test circuits are defined. In this case, DUT parameters that normally appear without a prefix in the DUT Parameters table will include the transistor name from the Test Circuit as a prefix. For the extractions to use these parameters, EXTR_DUT must be set to the transistor name that is used in the test circuit.

Extracting Parameters Through Optimization

It is not always possible to adequately isolate a circuit component before using a standard extraction function. In these cases it is still possible to extract model parameters by using the optimize function. As with any extraction function, successful use of the optimizer requires that the parameters being optimized have a dominant effect over the simulation of the

measured characteristics. Refer to Chapter 7, "Optimizing," in the *IC-CAP User's Guide* for more information regarding optimization.

In the OR/NOR gate shown in Figure 178, it is possible to use the optimizer to extract the values of NPN1.IS, NPN1.NF, NPN1.BF and RIEE. The following sequence of operations describes how to accomplish this.

1 Make an Ic and Ib versus V measurement between the NOR, IN1 and VEE terminals.

   a Connect the VEE, VREF and IN2 terminals to constant voltage sources of 0V. This keeps the base-emitter diodes of Q2 and Q0 in an off state. Disconnect VCC from the circuit.

   b Connect the NOR terminal to a voltage of approximately 1.0V.

   c Sweep the voltage on the IN1 terminal so that the measured currents at the NOR (collector) terminal are in the 1nA to 1µA range.

2 Set up an optimization transform that optimizes the values of NPN1.IS, NPN1.NF, and NPN1.BF over the measured current. At low currents, RIEE has a minimal affect on the I-V relationship.

   - The target data is the measured Ic and Ib currents. The simulated data comes from the simulation of these currents.
   - The Parameters table contains NPN1.IS, NPN1.NF and NPN1.BF

3 Change the sweep voltage on the IN1 terminal so that the measured current at the NOR terminal is in the 10µA to 1mA range.

   - The measured current should deviate from an exponential function due to the debiasing effect from RIEE.

4 Set up an optimization transform that optimizes the value of RIEE over the measured current.

   - The target data is the measured Ic current. The simulated data comes from the simulation of this current.

- The Parameters table contains only the circuit element RIEE.

After each of these measurements and optimizations has been executed, the Model Parameters table is updated with the extracted values of these elements.

Circuit Simulation

Circuit simulation is performed identically to single device simulation. The circuit usually has more inputs and outputs defined than a single device. In addition, the simulated circuit can use independent or controlled voltage and current sources that are defined within the Circuit Editor. When IC-CAP simulates a circuit, it first builds a complete SPICE deck from the circuit description and the setup table. The source names are built from the source type (V or I) and the nodes to which they are connected. Use of the simulation debugger can improve efficiency in performing successful simulations. Knowledge of how IC-CAP interacts with the SPICE simulators gives you more control over the options available for circuit simulation. For more information, refer to Chapter 3, "SPICE Simulators," in the *Reference*.

One of the advantages of simulating circuits through IC-CAP is the increased levels of analysis available. IC-CAP allows a sweep of more parameters than with a stand-alone SPICE simulator. For example, it is possible to simulate a circuit's behavior over bias conditions, component values and temperature in the same simulation. Once a simulation is complete, you may further analyze the stimulus and response data with IC-CAP's numerical and graphic capabilities. The two-port simulation features enable you to study the high-frequency characteristics of a circuit using any of the S, H, Y, or Z 2-port parameters.

Design Optimization

Designing a circuit usually follows a path of defining a block level functional description, translating it into discrete circuit components, then optimizing those components for the required performance. This last stage can be simplified with the IC-CAP system. It is possible to specify the desired performance from a

circuit and then let the IC-CAP optimizer find the best component and model parameters to satisfy it. The following is an overview of how to do this.

1. Enter a circuit with a first order estimate of the required parameters and component values.
2. Create a setup with the inputs and outputs that simulate the desired region of performance.
3. Simulate the circuit to create a set(s) of output data.
4. Copy the simulated data into the measured data set(s) in the outputs.
   a. Type the letter *S* in the *Type* field and press Return.
   b. Type the letter *B* in the same field to replace the letter *S*.
5. Save the desired outputs to files using the Write to File menu choice on each output.
6. Edit the files using any editor. Scan down the file to find the *type MEAS* data section. This is where the measured data begins. Edit the output values, replacing them with the desired performance values for the circuit. Save the file when done.
7. Read the file (and thus the new data) back into the outputs in IC-CAP using the *Read from file* command on the desired outputs.
8. Set up an optimization transform to find the required parameter and component values that best match the new measured data.

This type of design optimization can save many hours of iteration in fine-tuning high-performance circuit designs.

Test Circuits

When measuring a single device or a complex circuit you often require additional components for biasing, setting operating points, or tuning the high-frequency performance characteristics. Even when no additional components are required, there may still be some parasitic elements introduced by the connections of the device to the instrumentation. Examples of this are DC resistance in probe-to-wafer contacts, inductances in IC package bond wires, and the shunt resistances in the bias ports of network analyzers. The Test Circuit in IC-CAP allows you to include these elements when performing a simulation or optimization without including them in the main circuit description or device model.

Syntax

A circuit editor is available for each DUT. To access it, select Edit in the DUT Circuit folder. This produces a window that has both the DUT Parameters table and the test circuit editor. The mechanics of using this editor are the same as using the circuit editor.

The test circuit adds another level of circuit hierarchy to the overall system being measured and modeled. It is implemented through a circuit description that uses the SPICE subcircuit syntax. The example test circuit shown in Figure 179 adds a capacitor and resistor to the outputs of the ECL logic gate described earlier.

```
* Resistive / Capacitive circuits to
* simulate the effect of gate loading
.SUBCKT gateload 1=IN1  2=IN2  3=OR
+ 4=NOR  5=VCC  6=VEE  7=VREF
Xornor 1 2 3 4 5 6 7 ECLornor
Cor  3 0 1p
Ror  3 0 10MEG
Cnor 4 0 1p
Rnor 4 0 10MEG
.ENDS
```

Figure 179 Test Circuit for an ECL Logic Gate

This test circuit is added to the SPICE circuit deck each time a simulation is called or when an optimization that uses a SPICE simulation is performed. It does not modify the original model description in any way. The values of the elements in the test circuit can be modified in the DUT Model Parameters.

Hierarchical Modeling

The previous test circuit section on illustrated one way to include hierarchy in an overall circuit description. The test circuit, however, is at the highest level of hierarchy in an IC-CAP circuit. It is also possible to build a complete circuit by combining smaller circuit or transistor models into one subcircuit definition. This way, you can update the models of smaller subcircuits or individual components and have these changes automatically propagate into all circuits that use it.

Circuits Built from Sub-models

The ECL logic gate defined in Figure 178 uses two sizes of NPN transistors. Each transistor has a separate .MODEL card associated with it. These transistor model definitions can be removed from the logic gate circuit and reference other models currently active in IC-CAP. When a simulation is performed, IC-CAP includes these device models in the circuit definition.

To use external models, the models that you want to include must be in the IC-CAP model list. In the circuit definition that uses these model references, remove the .MODEL card. The model name used for the transistors should then be the same as the names in the IC-CAP model list. To use this technique for the ECL logic gate, read in models for the *NPN1* and *NPN2* transistors into IC-CAP. Then delete the two model cards in the logic gate circuit. The resulting model list and circuit description are shown in Figure 180 and Figure 181.

This approach provides flexibility. It allows you to keep a standard library of device models to include in larger circuits requiring accurate device models. This allows you to quickly *cut-and-paste* different component models into circuits and compare performance. It also greatly reduces the size of the circuit definition that needs to be maintained.

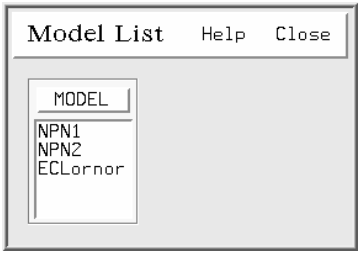


Figure 180 Model List Window for Hierarchical Circuit Definition

```
.SUBCKT ECLORNOR  1=IN1 2=IN2 3=OR 4=NOR
+ 5=VCC 6=VEE 7=VREF
* ECL OR/NOR LOGIC GATE
Q1   4  1  8  NPN1
Q2   4  2  8  NPN1
Q0   3  7  8  NPN2
RL1  5  4  300
RL0  5  3  300
RIEE 8  6  1.2K
.ENDS
```

Figure 181 Hierarchical Circuit Description for ECL OR/NOR Logic Gate

Functional Circuit Blocks

Previous sections in this chapter provided different aspects of the use of IC-CAP for modeling complete functional circuits. This section provides detailed examples of circuit models and custom model extractions you can create. These examples are provided to stimulate ideas for using IC-CAP to meet specific circuit modeling requirements.

Types of Circuits in IC-CAP

The types of circuits for measurement and simulation with IC-CAP are unlimited. Anything that can be simulated on a stand-alone SPICE simulator can be simulated with IC-CAP. In fact, any type of system that can be measured with the IC-CAP library of instruments can be characterized.

With the variety of components supported by SPICE, IC-CAP can be used to both characterize and design circuit modules. Classical examples of both single device and functional circuit modeling problems that are easily solved with IC-CAP are included.

Modeling the Reverse Active Region of an NPN Transistor

One of the constant sources of error in modeling the performance of a reverse active NPN transistor is the parasitic PNP formed by the base, collector, and substrate of the integrated structure. (This was briefly described in "PNP Transistors" on page 28.) A simple solution to modeling this region of operation is to use the complete functional circuit displayed in the following figure.

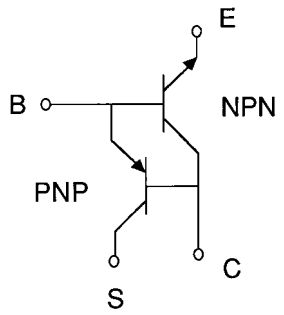


Figure 182 NPN Transistor with Parasitic PNP

Model file *npnwpnp.mdl* is included as an example of solving this problem. It has a single DUT/Setup that measure and model the reverse active operation of an NPN transistor.

The previous figure shows that the emitter of the PNP steals current from the base terminal of the NPN. The single dominant parameter that models this PNP current flow is the saturation current IS. (Modeling the transistor at this point assumes that the DC NPN parameters have already been obtained using another model file. The *bjt_npn.mdl* model file has a complete set of DUTs and setups to perform this. For more information, refer to Chapter 1, “Bipolar Transistor Characterization.”)

The *rgummel* setup then uses an optimize function to simultaneously extract the reverse active NPN parameters BR, IKR, ISC, and NC and the PNP parameter IS. The optimization proceeds by simulating the compound device, which is represented as a 2-transistor circuit, and numerically adjusting these model parameters.

The plot in the following figure is of reverse beta versus emitter current. It includes the simulation of the single transistor reverse model and measurement and simulation of the 2-transistor compound structure. The result of this extraction is a near perfect agreement between measured and simulated data. Also examine the resulting magnitudes of both IS (large enough to not be negligible) and BR (much higher than for a

single device extraction). This example shows the improvement you can attain in using a full circuit description to model an integrated device structure.

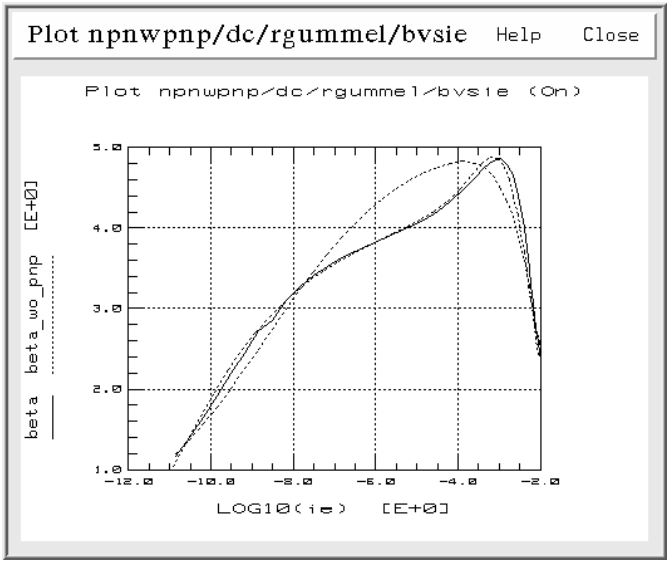


Figure 183 Reverse Beta versus Ie for Single NPN and Compound NPN-PNP Structure

Modeling an Operational Amplifier

The operational amplifier is included with IC-CAP as an example of how to do relatively complex macro modeling. It illustrates the simplification of a complex circuit to the necessary and sufficient components that enable it to be accurately represented. This model includes a Program transform that extracts model parameters from data sheet specifications of its performance. The inputs to this transform could also be measurements of the opamp's electrical characteristics. This same Program transform has also been converted into a standard IC-CAP function by writing it in the *C* programming language (in the *userc.c* module) and compiling

and linking it into the system. The circuit chosen follows the model developed by Boyle, Cohn, Pederson, and Solomon [1]. This circuit model is in the *opamp.mdl* example file.

Opamp Macro Model

The stages in this opamp model are: non-linear differential input, intermediate linear gain, and output driver. These enable most of the possible operating regions of the complete circuit to be adequately simulated.

The input stage contains transistors Q1 and Q2 connected as a differential amplifier, biased with a fixed current source (see Figure 183). Q1 and Q2 provide both differential mode (DM) and common mode (CM) characteristics. Passive components in the input stage provide slew rate effects (C2, Ce, Re1, Re2), 0dB frequency control (Rc1, Rc2), DM excess phase (C1), and CM input resistance (R1).

The intermediate gain stage provides linear amplification through voltage controlled current sources Ga, and Gb, which model the differential gain of the opamp. Capacitor C2 controls the dominant pole. CM gain is modeled with the voltage controlled source that has the coefficient Gcm.

In the output stage, AC and DC output resistances are modeled with Ro1 and Ro2. Output drive voltage is supplied through diodes D1 and D2 and voltage controlled voltage source (Gc*v6*Rc). The independent voltage sources in series with diodes D3 and D4 clamp the opamp under conditions that would force the output voltage to one of the supply rails.

The development of this opamp model uses the concepts of simplification and buildup to reduce the number of active and passive components. For example, to maintain non-linear effects the input stage has been simplified to two transistors, and an independent current source has been substituted for the usual bias circuitry. In the interstage amplifier, *buildup* is used to emulate a circuit characteristic through alternate circuitry. The stage is modeled by two voltage controlled current sources and a capacitor for compensation. These techniques take a piece-by-piece approach to the development of a model. They can be applied to virtually any circuit or subcell.

Opamp Circuit Model

The following figure shows the macro model that represents the full opamp circuit, followed by circuit elements in order from the input to the output stage. Figure 185 shows the complete opamp model circuit definition used in this example.

```
.SUBCKT OPAMP_1 2 = VINP 3 = VINN
+ 4 = VEE 6 = VOUT 7 = VCC
Q1 10 2 12 NPN1
Q2 11 3 13 NPN2
.MODEL NPN1 NPN IS = 8.0E-16 BF = 111.7
.MODEL NPN2 NPN IS = 8.31E-16 BF = 143.6
RC1 7 10 5305
RC2 7 11 5305
C1 10 11 5.46p
RE1 12 14 2712
RE2 13 14 2712
RE 14 0 9.872m
CE 14 0 2.41p
RP 7 4 15.36K
GCM 0 15 14 0 6.28n
GA 15 0 10 11 188.6u
R2 15 0 l23.4K
C2 15 16 30P
GB 16 0 15 0 247.5
RO2 16 0 42.87
D1 16 17 DMOD1
D2 17 16 DMOD1
.MODEL DMOD1 D IS = 8.0E-16
RC 17 0 0.02129m
GC 0 17 6 0 46964.0
RO1 16 6 32.13
D3 6 18 DMOD2
D4 19 6 DMOD2
.MODEL DMOD2 D IS = 8.0E-16
VC 7 18 1.803
VE 19 4 2.303
IEE 14 4 20.26u
.ENDS
```

Figure 184 Opamp Circuit Definition

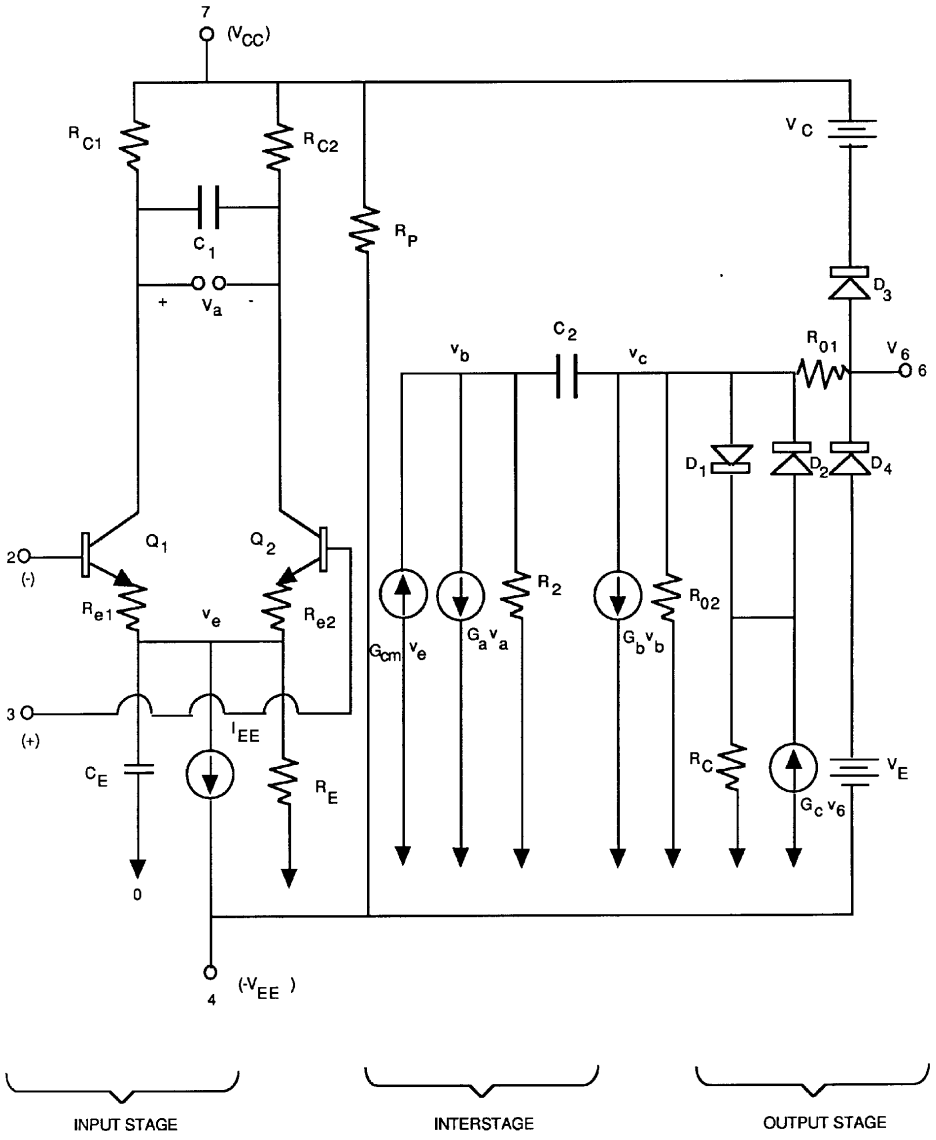


Figure 185 Opamp Circuit Diagram

Inputs to the Opamp Macro Extraction

Inputs to the opamp model extraction describe its electrical performance. These characteristics can be measured on actual devices or obtained from data sheet specifications. Due to the flexibility of the origin of the inputs, you can experiment with the opamp's performance as it relates to the model elements that control it. The following table lists the inputs to the opamp extraction.

Table 59 opamp Extraction Inputs

Input Name	Contents
Slew Rate +	positive-going slew rate
Slew Rate-	negative-going slew rate
Bias Current	average input base bias current
Bias Offset	input bias offset current
Volt Offset	input offset voltage
Av(DM)	open loop differential mode voltage gain
BW	unity gain bandwidth (Av(DM) * f(-3dB))
Excess Phase	excess phase at f(0dB) due to 2nd pole
CMRR (dB)	common mode rejection ratio
Rout	low-frequency output resistance
Rout-ac	high-frequency output resistance
Isc+	positive short circuit current
Isc-	negative short circuit current
Vout_max+	positive output voltage where opamp clamps Iout
Vout_min-	negative output voltage where opamp clamps Iout
Power Diss	quiescent state power dissipation
Vcc supply	positive supply voltage
Vee supply	negative supply voltage
Nom. Q.IS	nominal transistor and diode saturation current

Table 59 opamp Extraction Inputs (continued) (continued)

Input Name	Contents
R2	differential gain setting resistor
Comp. Cap.	compensation capacitance
Temp.(C)	temperature for macro extraction and input specification
Inputs PNP?	flag to switch the inputs to PNP transistors
Debug?	flag to turn on debug output during extraction

Extraction Equations for the Opamp Macro Model

The inputs to the macro model extraction described are used by the set of equations shown in Figure 186 to produce the model parameters. These equations are programmed into the *userc.c* module exactly as shown.

Because of the simplicity of the equations, they can also be entered into a Program transform using the Parameter Extraction Language. This allows experimentation with model extraction techniques before coding the final extraction in C and linking it to IC-CAP. This has been done with the opamp macro model to provide a typical example of writing custom model extractions.

```
if ( is_pnp )
/* for pnp, pos. and neg. slew-rates are interchanged in calculations */
     { tmp = sr_plus ; sr_plus = sr_neg ; sr_neg = tmp ; }
vt = 25.85E-3 * (temp/27);
isc = ( isc_p + isc_n ) / 2 ;
is1 = isd3 = isd4 = is ;
sr_plus * = 1.0E6; /* convert slew rates to Volts/sec */
sr_neg * = 1.0E6;
ic1 = ic2 = (c2/2) * sr_plus ;
ce = 2*ic1/sr_neg - c2 ;
ib2 = ib + ibos/2 ; /* ib2-ib1 = ib+ - ib- = ibos */
ib1 = ib - ibos/2 ;
if (is_pnp)
/* bias currents will have opposite signs with pnp input stage */
     { ib1 = -ib1 ; ib2 = -ib2 ; }
beta1 = ic1/ib1 ;
beta2 = ic2/ib2 ;
iee = (( beta1+1 )/beta1 + (beta2+1)/beta2 ) * ic1 ;
re = 200/iee ;
is2 = is1 * ( 1 + vos/vt ) ;
gm1 = ic1/vt ;
rc2 = rc1 = 1 / ( 2*M_PI*fOdB*c2 ) ;
re1 = re2 = (beta1+beta2) / (beta1+beta2+2) * (rc1 - 1/gm1) ;
c1 = c2 / 2 * tan(delta_phi) ;
/* Vcc is defined as being a positive value & Vee as a negative value */
rp = vcc-vee) * (vcc-vee) / (pd - fabs(vcc*2*ic1) - fabs(vee * iee) ) ;
ga = 1/rc1;
cmrr = pow(10.0,cmrr/20.0); /* convert from dB to decimal */
gcm = 1 / (rc1*cmrr) ;
ro1 = rout_ac ;
ro2 = rout - ro1 ;
gb = avd * rc1 / ( r2 * ro2 ) ;
ix = ( 2 * ic1) * gb * r2 - isc ;
isd1 = isd2 = ix * exp( -ro1*isc/vt ) ;
rc = vt / (100*ix) * log ( ix/isd1 ) ;
gc= 1/rc;
vc = fabs(vcc) - vout_p + vt*log(isc_p/isd3) ;
ve = fabs(vee) + vout_n + vt*log(isc_n/isd4) ;
```

Figure 186 C Coded Opamp Extraction Equations

Bias Circuitry

The opamp model, fully defined and model parameters extracted, can now be used as a functional circuit block. This requires that the opamp be biased with external supply voltages and circuit configuring components. This was demonstrated previously through the use of a test circuit. The test circuit implemented forms an inverting amplifier using an input and feedback resistor with the opamp. This results in a complete functional circuit whose performance can be studied in more detail. The test circuit definition and the resulting equivalent circuit are shown in Figure 187 and Figure 188.

```
* Inverting Amplifier
.SUBCKT  inv_amp 1 2 3 4 6 7
X1   2 3 4 6 7  opamp_1
Rf     6  2  10K
Rin    2  1  2K
Rgnd   3  0  1.0m
.ENDS
```

Figure 187 Test Circuit Definition to Form an Inverting Amplifier

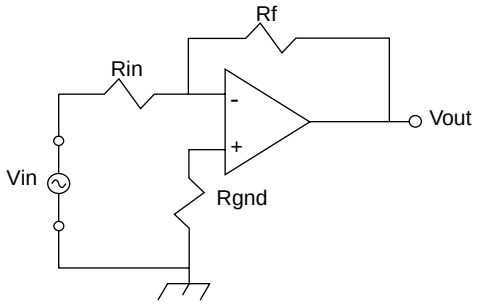


Figure 188 Equivalent Schematic of Opamp in Inverting Amplifier

The plot in Figure 189 illustrates how this functional circuit can be studied. The opamp circuit has been simulated with the compensation capacitance used as one of the sweep parameters. The AC voltage gain is plotted versus frequency with steps of different values of C2. The diagonal line is the break point between process limitations and circuit limitations. To increase frequency response, the internal capacitor on the chip (nominal 30pF) would have to be reduced. To force earlier roll-off, more external capacitance can be added. This analysis indicates the range of gain-bandwidth product available for different values of capacitance. Similar analyses can be performed for any area of this circuit's operation.

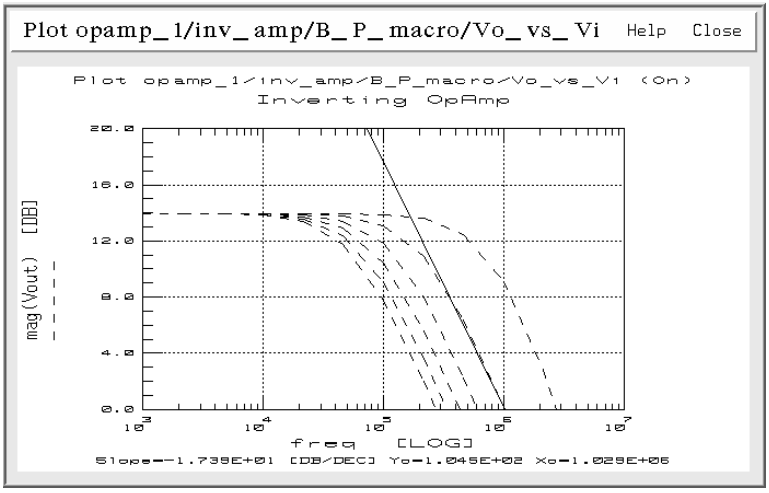


Figure 189 AC Opamp Response vs C2 Capacitance

References

1 G.R. Boyle, B.M. Cohn, D.O.Pederson, J.E. Solomon. *Macromodeling of Integrated Circuit Operational Amplifiers,* IEEE Journal of Solid-State Circuits, Vol. SC-9, No. 6, December 1974.

10
1/f Noise Extraction Toolkit

Types of Noise 568
The Significance of 1/f Noise 569
1/f (Flicker Noise) Modeling 570
Noise Measurement Setup 573
Parameter Extraction Procedure 575
1/f Noise Toolkit Description 578
Opening the 1/f Toolkit 579
System Settings 582
General Settings 584
Verify DC 590
Measure Noise 595
1/f Noise Extraction with MOS Transistors 614

The 1/f Noise Extraction Toolkit runs with a GUI. The Toolkit can measure and extract 1/f noise parameters.

Types of Noise

The main sources of noise in semiconductor devices are:

1. **Thermal Noise** is due to the Johnson effect in the ohmic regions. It is practically frequency independent.
2. **Shot Noise** is due the diffusion and passage of carriers across barriers. This noise can be represented by independent random events. Its spectral distribution depends on the bias current and is frequency independent.
3. **Burst Noise** is due to the capture and emission of carriers in localized traps causing a fluctuation between current levels. It is usually present in small devices and Si-SiO2 interfaces (e.g. MOS).
4. **1/f Noise** (also called Flicker Noise) is believed to be caused by surface recombination due to traps and defects in the crystal.

The Significance of 1/f Noise

Due to their broadband spectral distribution, thermal and shot noise contribute to the circuit noise figure. The circuit (in terms of bias, gain, matching networks, etc.) is usually optimized to minimize or lower the noise figure.

Although the 1/f noise is generated at low frequencies, its contribution to the overall noise can be very significant to some RF circuits since its spectral power density is up converted in the band of interest by circuit nonlinearities. This is especially true in mixers and oscillators.

1/f (Flicker Noise) Modeling

The state of the art of 1/f noise modeling will be reviewed, and then the 1/f noise setup will be described with more details. The 1/f noise spectral power density is given by [8]:

$$S_{1/f}(f) = K_f \cdot \frac{I^{A_f}}{f} \tag{128}$$

where Kf and Af are the model parameters that needs to be extracted, I is the current flowing through the junction where the flicker noise is generated. The figure below shows the small signal equivalent circuits of bipolar and MOS transistors. All the noise sources, including thermal and shot, are represented. In bipolar transistors the noise is generated by the recombination current in the base, therefore $I = I_b$. The 1/f noise is represented by the current source Inb. Its squared mean value is given by:

$$\langle I_{nb}^2 \rangle = \oint_B S_{1/f}(f)\,df \tag{129}$$

where B is the frequency band of interest.

In FET and MOS devices the noise is generated in the channel, therefore $I = I_b$. In the MOS equivalent circuit shown below, the 1/f noise is represented by the current source Ind. The expression for the density is:

$$S_{1/f}(f) = K_f \cdot \frac{I_d^{Af}}{f^{Ef} \cdot C_{ox} \cdot L_{eff}^2} \tag{130}$$

The bipolar equivalent circuit used is shown below.

The BSIM3 and BSIM4 models for MOSFETs offer, as an alternative, a more complex description which includes other device parameters. As with the conventional BSIM3 and BSIM4 formulation in equation (3), some models (e.g., VBIC) also offer a parameter Ef to model the 1/f slope. The goal is to be able to measure the noise spectrum at low frequencies, where the 1/f noise is dominant, and at several bias points in order to extract the parameters Kf and Af.

Kf is determined by the power vs. frequency characteristics; Af is extracted by varying the bias current.

In both cases, bipolar or MOS, the noise phenomena is related to the current flowing through the device. The formulation gives the spectral density of a random noise current source. The bipolar transistor is by nature a current amplifier. The equivalent circuit is shown above.

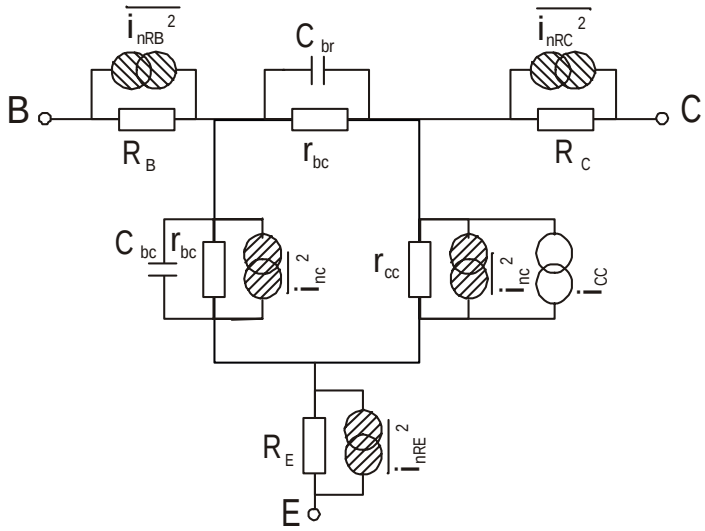


Figure 190 Equivalent Circuit of a MOS Transistor

The power noise spectrum of the base current is simply amplified by a factor h^2_{fe} at the output (h_{fe} is the small signal current gain). Having considered this, the most natural choice for amplifying the device output noise is a current amplifier.

The low noise current amplifier converts the output current noise into voltage. This avoids current to voltage conversions in the circuit. These conversions depend on other circuit parameters, and therefore introduce errors.

Noise Measurement Setup

The figure below shows the block diagram of the measurement setup. The input bias is applied by a 4142 (or 4156) parametric analyzer and a 1Hz or a 10 Hz filter. The filter eliminates the line noise from the bias source. When measuring bipolar devices, the filter output impedance, which is typically 50 ohm, has to be increased since such a low value would short circuit the current noise source at the input.

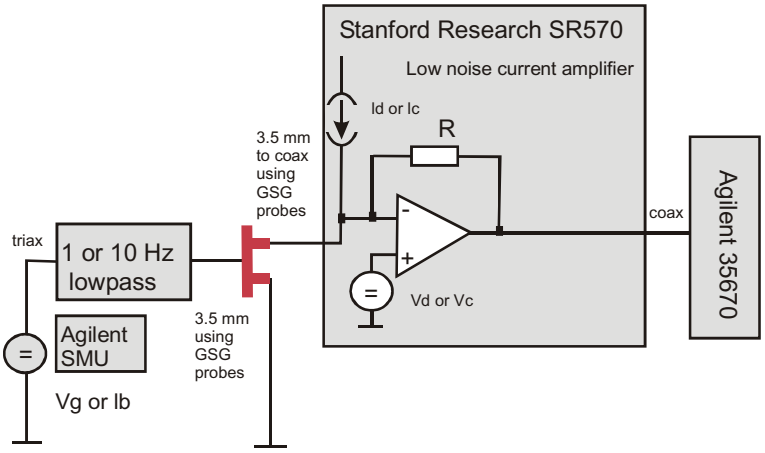


Figure 191 1/f Noise Measurement Setup

The device output is directly connected to the SR570 low noise amplifier. Besides amplifying the output noise, the amplifier provides a current and a voltage supply which are used to bias the device output. The voltage supply allows the setting of the collector or drain voltage while the current source provides an offset current which is used to bias the device. The reason for using the current compensation is that the amplifier works best when the feedback current is minimal (the feedback current flows through R into the device to create the virtual null at the amplifier input). The SR570 can supply an output voltage of up to 5 V and a maximum current compensation of 5 mA.

The gain (expressed in terms of sensitivity A/V) can be varied between 10e-3 to 10e-12 A/V. For this application, since the amplifier bandwidth decreases with the sensitivity, only the higher sensitivities which ensure bandwidth of at least 1 KHz are actually used.

The device 1/f noise is amplified and measured by the 35670A dynamic signal analyzer. This instrument is ideal for analyzing signals with a low frequency power spectrum (such as 1/f noise) as opposed to a spectrum analyzer which is used at higher frequency bands. Because of the relatively low frequency range, the dynamic signal analyzer can directly sample the signal over a 1/f period and operate a FFT transform to calculate the spectral density.

Parameter Extraction Procedure

As mentioned already, to extract the noise parameters, Af and Kf, the 1/f noise spectral density of the noise current source needs to be measured at various bias points. Since the current amplifier does not have GP-IB control, its settings (bias voltage, current offset and sensitivity) have to be manually set at each bias point. The measurement is therefore semi-automated with the data acquisition and display controlled by IC-CAP. IC-CAP provides different model files for bipolar and MOS 1/f noise, but similar extraction procedures. The procedure consists of three steps and a verification phase.

In the BJT extraction, the Ic vs. Vbe DC traces are measured. This is accomplished by running the step Measure and Simulate DC Data in the GUI sequence. The DC current gain β, and the output conductance g_{ce} are calculated using the measured data.

In the MOS extraction, the Id vs. Vd DC traces are measured instead and the output conductance g_o is calculated.

Based on those data the user is asked to choose the bias points where the 1/f noise will be measured. Running the macro GUI step Measure Noise Data starts the interactive procedure for measuring the noise. For each bias point, the device is first biased at the base after some time that depends on the filter time constant. The operator is then asked to set the collector voltage, current offset and sensitivity on the SR570. The toolkit then calls the dynamic signal analyzer for measurement of the output noise. The spectral noise density at the output of the device is given by:

$$S_{ic}\langle f\rangle = (N_{meas} \times S_{SR570})^2 \quad \textbf{(131)}$$

where Nmeas is the noise measured by the analyzer expressed in $V/\sqrt{H_z}$ and SR570 is the sensitivity of the amplifier. In the bipolar case, assuming the DC beta is equal to the small signal current gain, the actual spectral power density of the base current noise source is given by:

$$S_{ib}\langle f\rangle = S_{ic}\langle f\rangle / \beta_{tr}^2 \quad \textbf{(132)}$$

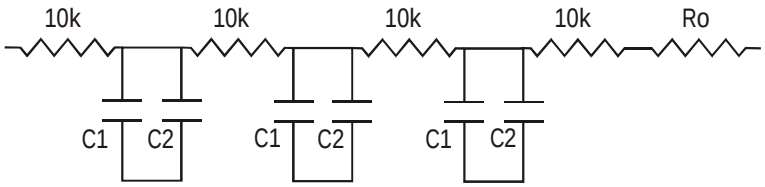
The third step is the extraction of the 1/f noise parameters. In the bipolar case, after selecting a set of traces at constant Vc, a macro linearizes equation (1) as follows:

$$\log(fS_{ib}(f)) = \log(K_f) + A_f \log(I_b)$$

and extracts the Af and Kf using a linear interpolation in the area where 1/f noise is present. Typically, the extraction is performed between 10 and 100 Hz of the 1/f band.

Low pass filter

The filter schematic is shown in the figure below:



C1 = 100 µF electolytic
C2 = 100 pF ceramic
R0 = 50 Ω for CMOS
Ro = 330 kΩ for bipolar

For Noise measurements of bipolar transistors, the low pass filter can be designed using an RC values network with a large output impedance (Ro = 330 kOhm). This large resistance prevents the noise voltage across the base of the transistor from being shorted out. Due to these large value, the filter series resistance is rather large, and so is the RC constant. This large resistance slows down the filter charge, and the device bias time is in the order of minutes. Also, the voltage across the filter might be several volts, therefore the input SMU voltage compliance must be set to a rather high value.

For Noise measurements of MOS transistors, the resistance Ro might be set to 50 ohm. The series resistances of the filter is 30k ohm, so that the filter charging will not be as long as in the bipolar case.

1/f Noise Toolkit Description

- The Toolkit uses IC-CAP .mdm model files.
- The GUI layer uses advanced data management techniques for handling .mdm model files.
- New GUI interface leads the user through DC and Noise measurements and extraction.
- Dedicated transforms drive the 4142 or 4156.

System Parts List (Hardware)

- Stanford Research low noise amplifier LNA SR570.
- Agilent Dynamic Signal Analyzer (DSA) 35670A.
- Agilent 4142B or 4156B/C.
- Voltage DMM.
- 1 or 10 Hz low pass filter.
- Cables (BNC, triax).
- GSG probes for on-wafer measurement.

Software Requirements

- IC-CAP software. Agilent 85190A, revision 2001 or higher.
- DSA Driver. Agilent 3567A driver is the 85199G.
- IC-CAP Model Files (part of the new toolkit). These files are located in the directory examples/model_files/noise/1_f_toolkit.
- 1/f Noise Toolkit License for running Measurement and Extraction of these files.

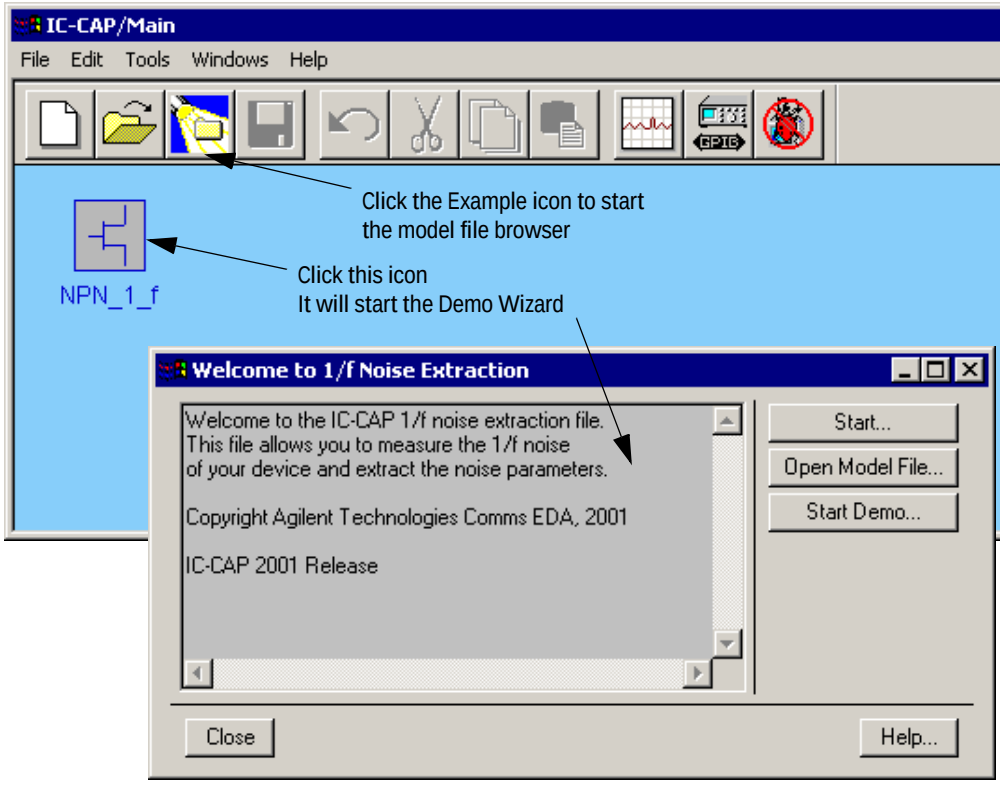
Opening the 1/f Toolkit

In the IC- CAP/Main window click on the Examples icon to open a browser.

See figure below. Select:
.../examples/model_files/noise/1_f_toolkit/bjt_1f_noise.mdl

Click on the NPN_1_f icon to start the Demo Wizard.

NOTE The step by step explanations refer to bipolar junction transistor (BJT) technology. MOS will be described in a later section.


IC-CAP/Main
File Edit Tools Windows Help
Click the Example icon to start
the model file browser
NPN_1_f
Click this icon
It will start the Demo Wizard
Welcome to 1/f Noise Extraction
Welcome to the IC-CAP 1/f noise extraction file.
This file allows you to measure the 1/f noise
of your device and extract the noise parameters.
Copyright Agilent Technologies Comms EDA, 2001
IC-CAP 2001 Release
Start...
Open Model File...
Start Demo...
Close
Help...

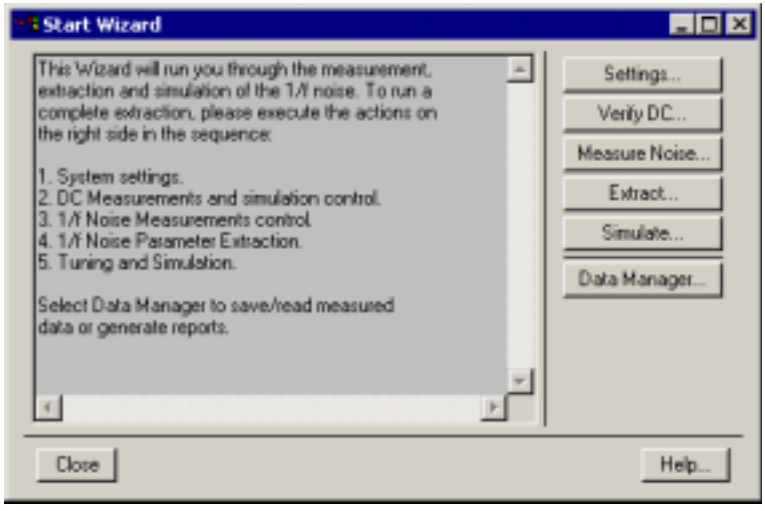
This Wizard will run you through the measurement, extraction and simulation of the 1/f noise. To run a complete extraction, please select these buttons in sequence.

On the wizard's window labeled Welcome to 1/f Noise Extraction there are three buttons: Start, Open Model File, and Start Demo.

- Click Start to start measurement setting, perform DC and noise measurements and extract 1/f noise parameters.
- Click Open Model File to edit variables, DUTs, setups etc.
- Click Start Demo to begin a demonstration of the measurement and extraction procedure (Measurements cannot actually be performed during the demo).

Click Start on the Welcome to 1/f Noise Extraction window. It will launch the Start Wizard window shown below.

During a DEMO the label Demo Mode Active will appear on all dialog boxes to remind you that the system is not in the full measurement mode.


Start Wizard
This Wizard will run you through the measurement, extraction and simulation of the 1/f noise. To run a complete extraction, please execute the actions on the right side in the sequence:
1. System settings.
2. DC Measurements and simulation control.
3. 1/f Noise Measurements control.
4. 1/f Noise Parameter Extraction.
5. Tuning and Simulation.
Select Data Manager to save/read measured data or generate reports.
Settings...
Verify DC...
Measure Noise...
Extract...
Simulate...
Data Manager...
Close
Help...

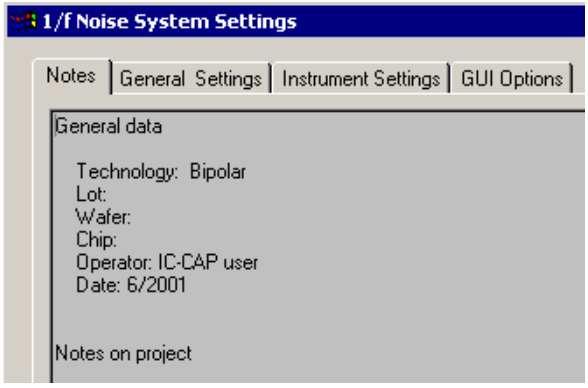
This window let you access the various steps of the measurement and extraction procedure.

- - Click on **Settings** to access system settings (file notes and comments, bias settings, system hardware configuration and instrument option settings).
- - Click on **Verify DC** to open the DC measurements wizard.
- - Click on **Measure Noise** to open the Noise measurements wizard.
- - Click on **Extract** to extract the noise parameters from selected measured data.
- - Click on **Simulate** to compare measured and simulated noise and tune the extracted parameters.
- - Click on **Data Manager** to read/save data and generate reports.

System Settings

Start > Settings

Clicking on Settings in the Start Wizard window launches the 1/f Noise System Settings window shown below.



1/f Noise System Settings

Start > Settings

Settings launches a dialog containing tabbed folders for 4 categories:

- Notes
- General Settings
- Instrument Settings
- GUI Options

Each of these categories uses a tabbed folder in the 1/f Noise System Settings window.

Notes

Start > Settings > Notes

The Notes page allows any free-form entry, typically:

- the technology,
- the source of the wafer, chip or package used for the test,
- the operator,
- data of special interest to this measurement,
- etc.

NOTE

Any text entered on the Notes page will be printed in the final report.

In the Demo Mode the user is not allowed to change the input data (SMU settings) since this action would clear the measured data.

General Settings

Start > Settings > General Settings

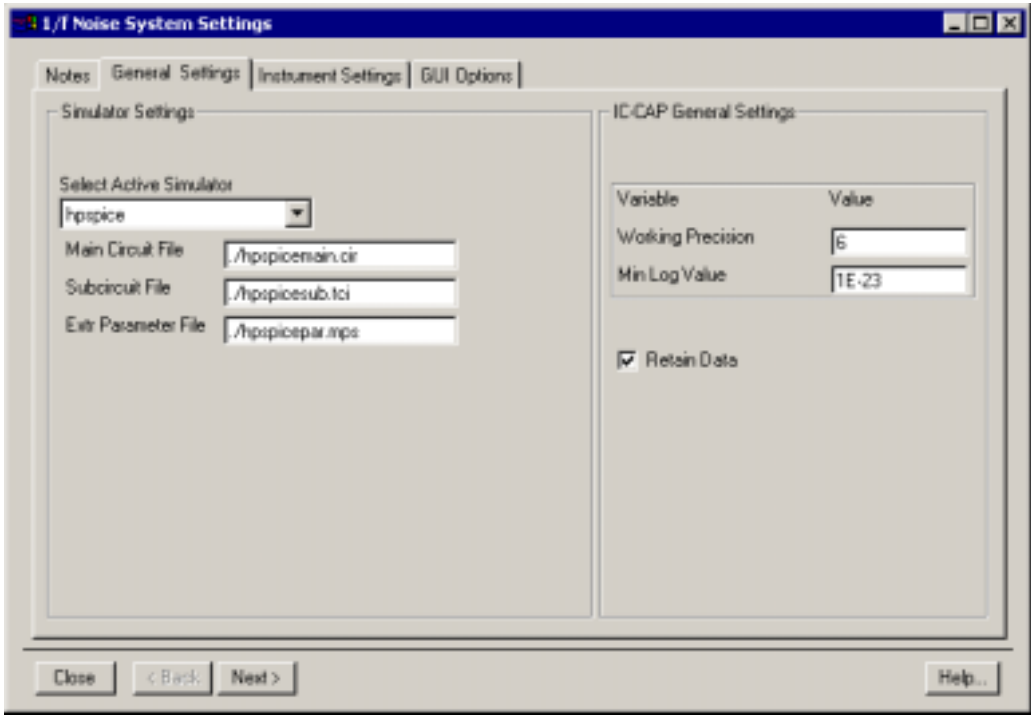
The Simulator Settings part of the dialog box allows

- Selection of a simulator, for example, hpspice or spectre.
- Specifying the Main Circuit File, Subcircuit File, and Parameter File. These are automatically loaded into the model file

CAUTION Changing the simulator will load new Circuit, Subcircuit, and Parameter files (if those are found). Previous data will be lost.

The IC-CAP General Settings part of the dialog box allows setting:

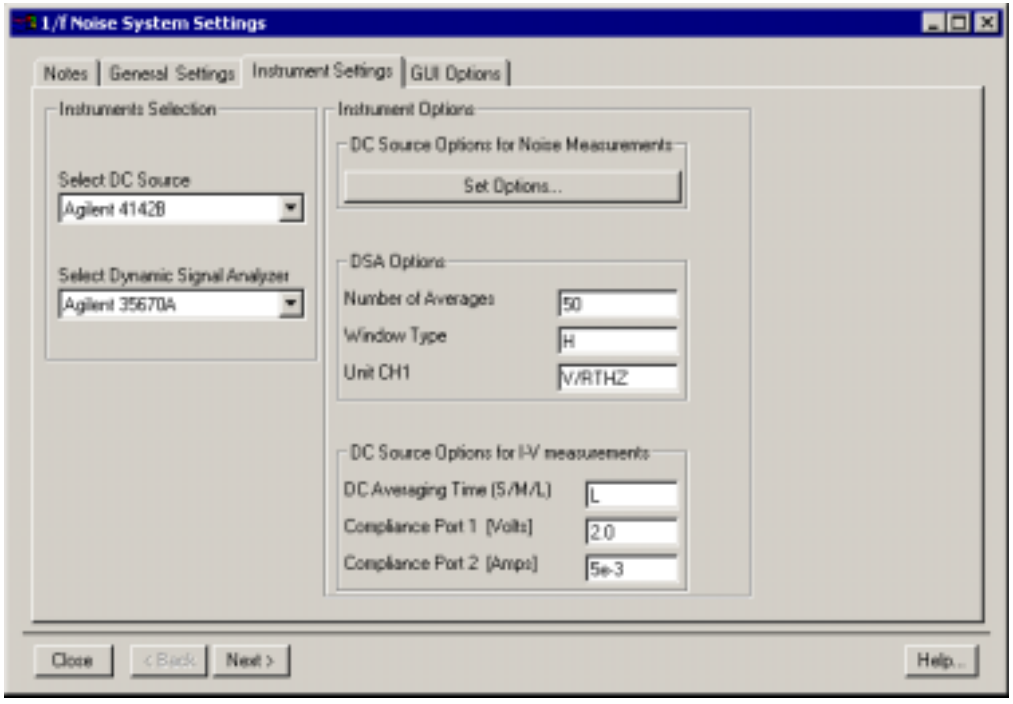
- **Working Precision:** this value must be at least set to 6, since the NDA frequency range must be saved with at least 6 figures.
- **Min Log Value:** defines the value to be used in a LOG plot, if data point value is zero or negative. Default is 1e–18 but for this application needs to be lowered to 1e–23.
- **Retain Data**: causes data from a Setup to be retained if a sweep changes but the number of points remains the same. Default in this application is Yes, which causes the data to be kept.



Instrument Settings

Start > Settings > Instrument Settings

Click on **Instrument Settings**. The the tabbed page, shown below, appears.



The Instruments Selection part of the dialog box allows

- Selection of the DC Source for noise measurements - e.g. Agilent 4142B or Agilent 4156B/C
- Selection of the Dynamic Signal Analyzer, e.g. Agilent 35670A

NOTE

The selection of the DC Source for the Verification DC Measurements (I-V curves) is done in the IC-CAP Hardware setup window since the IC-CAP driver for the 4142 or 4156 is used.

At present only the Agilent 35670 is supported. However you can use another analyzer, and modify the model file accordingly.

The Instruments Options part of the dialog box allows

1 Setting DC Source Options for Noise Measurements by clicking on Set Options...

- Interface Name, e.g. lan[ ]:hpib, hpib, /dev/gpib0
- DC Source Address GP-IB address of the DC Source
- Unit Slot Number (For the 4142 or 4145, this is the actual slot of the SMU being used to bias the input of the device.)
- Force I/V For bipolar, current (I) is typically forced into the base. For MOS, voltage (V) is typically forced on the gate.
- Compliance Max voltage when forcing current (keep in mind the series resistance of the filter). Max current when forcing voltage.

These 5 settings (above) are used by the transforms NOISE_1f_force_ bias and NOISE_1f_stop_ bias to force the bias during noise measurements.

2 Setting DSA Options (These settings refer to the Agilent 35670 DSA)

- Number of Averages
- Window Type
- UnitCH1

3 Setting DC Source Options for I-V Measurements

The Toolkit uses the internal driver for the 3570 to measure the spectral noise density of the output noise. These settings will change the instrument options in the setup /Measure/Noise.

These settings are used by the 4142 or 4156 IC-CAP drivers to measure the I-V curves. See the Verify DC dialog.

- DC Averaging Time (S/M/L)
- Compliance Port 1
- Compliance Port 2

These settings will change the Instrument Option settings in the setup /Measure/DC-Sweep.

GUI Options

Start > Settings > GUI Options

The General GUI Options part of the dialog box allows

- General GUI Options
  - Activate check boxes for LNA settings

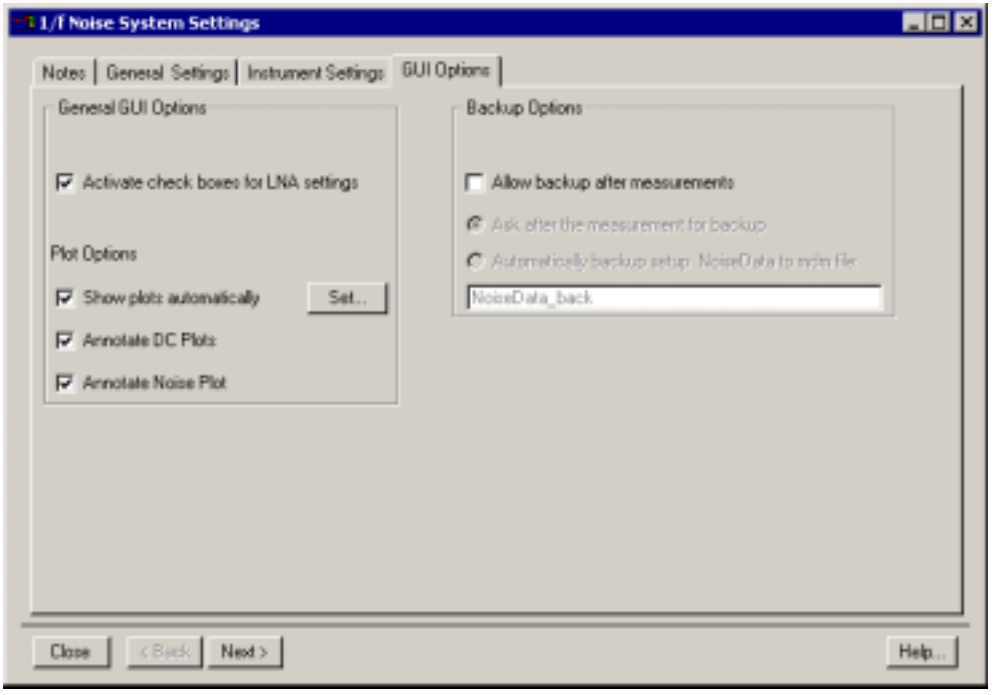
  This option will display 3 check boxes - to be checked at each Noise Point measurement. The operation will go through each step of the settings each time checking the corresponding box. Only when all the boxes are checked will the measurement be allowed to proceed.

Plot Options

- Show Plots Automatically
- Annotate DC Plots
- Annotate Noise Plot

The Backup Options part of the dialog box allows

- Allow backup after measurements
- Ask after the measurement for backup
- Automatically backup setup Noise Data after measurements to mdm file: <filename>.

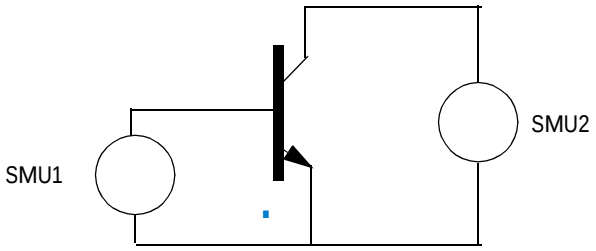

1/f Noise System Settings
Notes
General Settings
Instrument Settings
GUI Options
General GUI Options
Activate check boxes for LNA settings
Plot Options
Show plots automatically
Set...
Annotate DC Plots
Annotate Noise Plot
Backup Options
Allow backup after measurements
Ask after the measurement for backup
Automatically backup setup NoiseData to m/m file
NoiseData_back
Close
< Back
Next >
Help...

Verify DC

Start > Verify DC

Clicking on Verify DC in the Start Wizard window launches the Measure and Simulate DC Data window, shown below in two parts. The objective of this measurement is to verify the transistor I-V curves, and (for bipolar) tune the DC current gain, β, and calculate the output conductance g_0.

This window guides you through the DC measurements and simulation of the Ic-Vce characteristics.



DC Measurement Test Setup

Buttons on the Left Side of the Dialog:

- Run All: Run measurement and simulation automatically.
- Measure: Run measurement in the setup DC_sweep.
- Simulate: Run simulation in the setup DC_Sweep.
- Display All: Display all the DC plots.
- Close All: Close all the DC plots.

The DC working area consists of three parts:

- DC Verification Measurements Set the DC bias source (start, stop number of points) for input and outputs. Set integration time. Run the measurement or clear the measurement data.

- Plots A text comment region lets you write text which will be reported in the output characteristics plot. Use Update Plot Annotation to update plot comments.The annotation macro is defined under Macros and is called dc_plot. Annotation will be printed with the plot. Also, any annotation text will be printed on the final report.
- Simulation Select the simulator. Use the Simulate button to run the simulation. The Tuning table allows you to fine tuning the device current gain β. This is extremely important since the device β will be used later in the final tuning of the extracted parameters obtained by comparing measured and simulated noise.

CAUTION Changing the simulator will load new Circuit, Subcircuit, and Parameter files (if those are found). Previous Circuit/Subcircuit and extracted parameters will be lost.

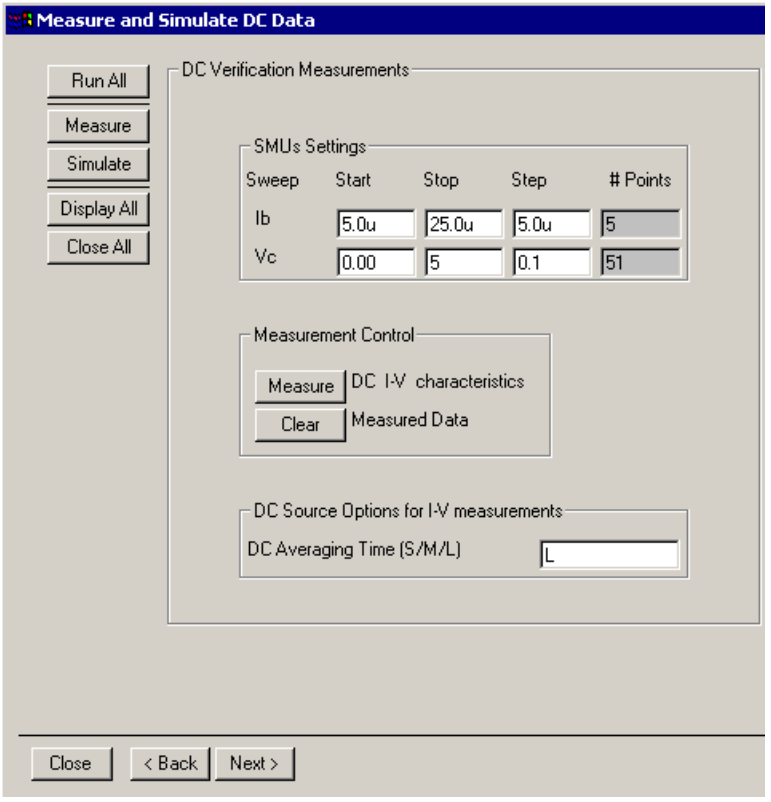


Figure 192 The Measure and Simulate DC Data Window - LEFT HALF

The left side of the dialog box is captioned DC Verification Measurements. It is used to make automatic measurements of a number of points. This number is determined by the product of the number of Ib points swept and the number of Vc points swept. The DC Averaging Time option changes the DC Analyzer integration time without re-opening the Settings dialog. An internal transform checks start, stop, and step input values. When errors are found, the input field turns red, and a warning is written to the IC-CAP status window. The transform also checks whether the total number of DC points for DC I-V measurements is greater than the maximum number allowed (see Settings).

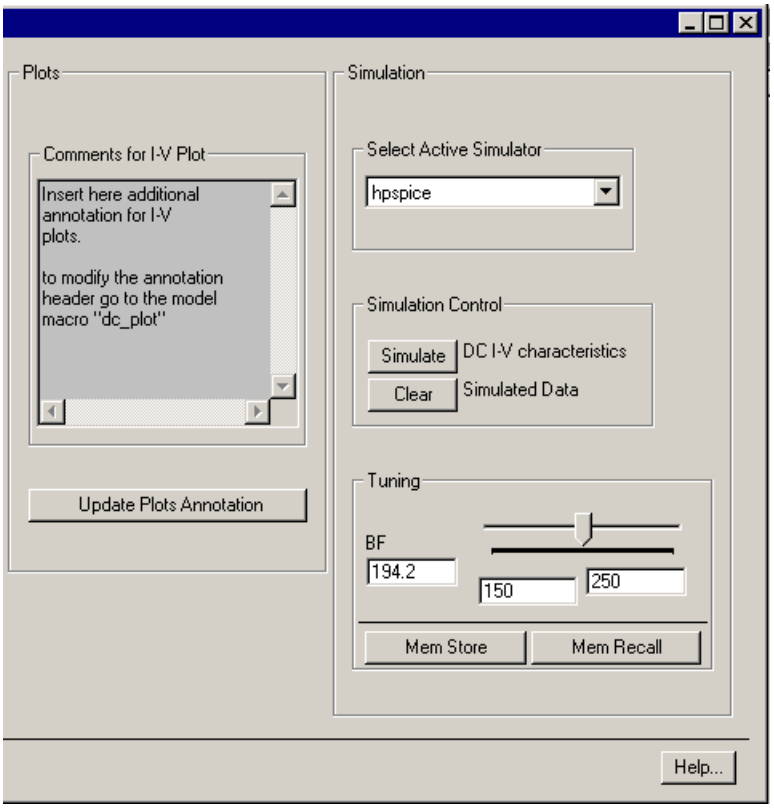


Figure 193 The Measure and Simulate DC Data Window - RIGHT HALF

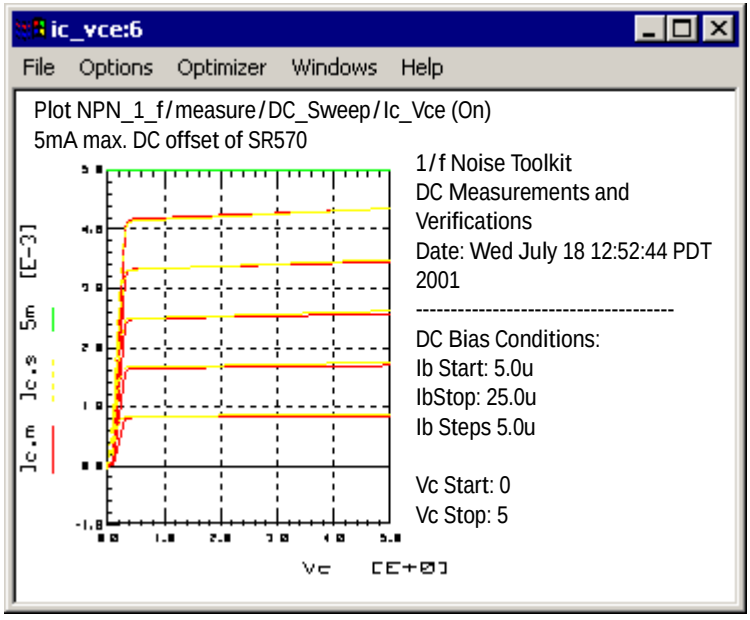
CAUTION Changing the simulator at this point forces a reload of the circuit, subcircuit, and parameter decks. Any previous changes will be lost unless they are saved first.

Tuning Beta

In order to simulate noise values you need a value of beta that is a good fit over the measured range of data. This is done by setting the maximum and minimum values of beta - in this example 150 and 250 respectively (see previous figure.). The actual value used is controlled by the slider. In the Ic versus Vc

plot (titled ic_vce), the yellow simulated (calculated) curves yellow match the red measured data curves best with beta (BF) at 150. See Figure below.

The Mem Store and Mem Recall buttons in the Tuning box allow you to store and recall the “tuned” values of beta for the subsequent noise measurements.



Measure Noise

Start > Measure Noise

The diagram below shows the general setup for noise measurement.

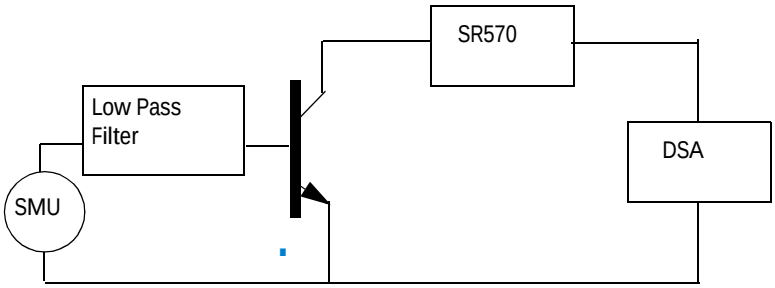


Figure 194 Noise Measurement Test Setup

Clicking on Measure Noise on the Start Wizard window launches the Measure Noise Data window shown below.

This half of the dialog box is captioned Noise Measurements (All Points). It is used to make automatic measurements of a number of DC bias points. This number is determined by the product of the number of Ib points swept and the number of Vc points swept. This is typically 10 to 20—in this example 15. Within this box is the Noise Bias SMU Settings box. You use this to set the start and stop values of Ib and Vc, and the step sizes. From this data, the number of points in each range is calculated and displayed, and also the total number of Noise Bias Points.

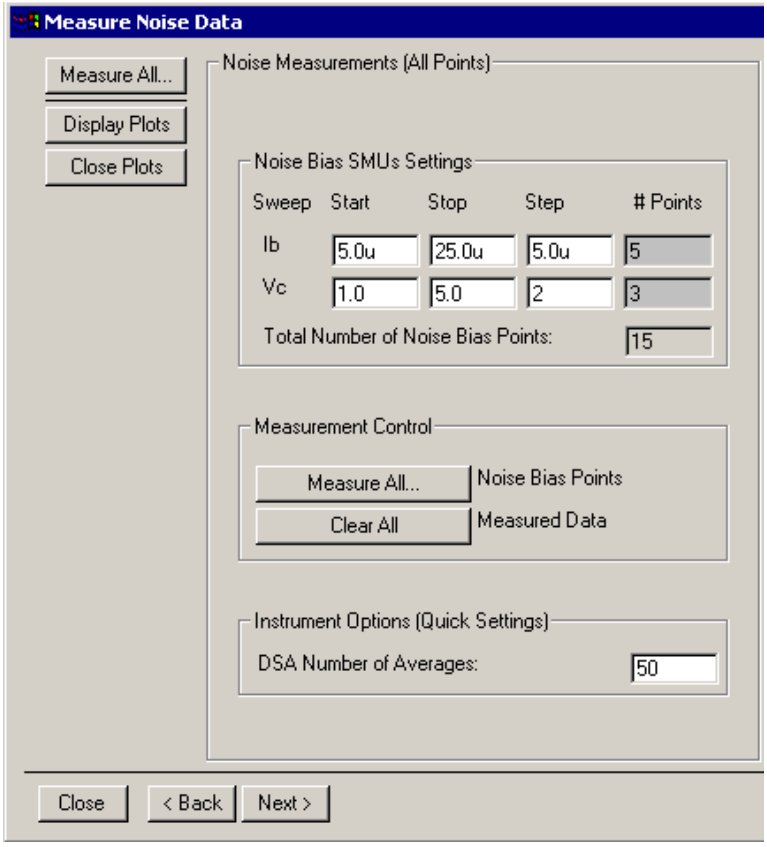


Figure 195 The Measure Noise Data Window - LEFT HALF

NOTE The Noise Points *MUST* be a subset of the points measured in the Verify DC setup.

In the Measurement Control box the Measure All button is used to start the automatic measurement.

The Clear All button is simply a convenience in starting a new measurement setup.

The Instrument Options box contains only the DSA Number of Averages value. This is typically between 50 and 100.

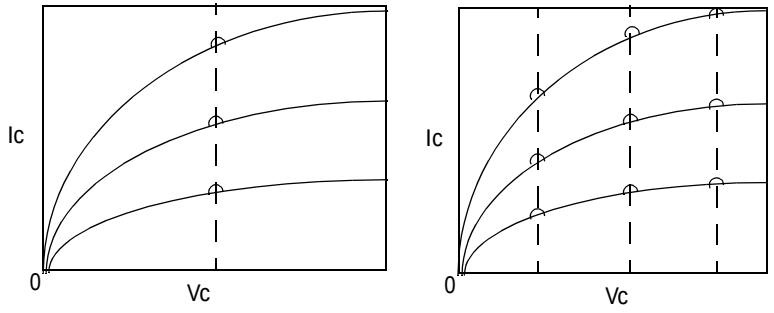
The button titled Display Plot only displays noise measurements at constant Vc.

Individual Noise Measurements at constant Vc

The box captioned Noise Measurements at constant Vc is used to make noise measurements at constant Vc only. The bias conditions are chosen from the set defined in the All Points window to the left.

Use the Choose Vc dropdown list to select the value.

NOTE When selecting a Vc value, the previously measured noise points at other Vc values are skipped during this measurement, and the past measurement data are left unchanged in the repository.



Measurement at a Single Constant Vc

Measurement at All Points

Figure 196 Measurement of Ic at a Single Constant Vc versus at All Points

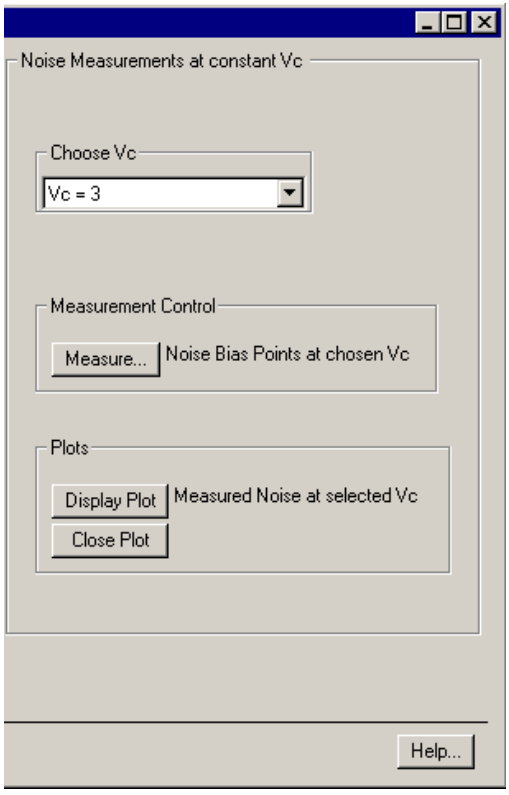


Figure 197 The Measure Noise Data Window - RIGHT HALF

Use the Measure button to start the measurement. It will launch a window titled Measure Single Noise Point Setup shown in the next figure.

This dialog gives you guidance and a checklist for completing the single point measurement. The top box titled Now Measuring gives the bias point sequence number, and the Vc an Ib bias conditions. The next four boxes let you check and launch the measurement.

NOTE Before launching the Single Measurement Point dialog, IC-CAP will automatically call the transform set_bias_force, which sets the Ib current into the base. At that point the input filter starts charging. Due to the low current level and the high RC constant, the charging can take as long as few minutes. In the meantime, user can proceed to step 1.

Step 1 Set the output bias voltage on the Low Noise Amplifier (LNA). In the Stanford Research LNA this has to be done manually. In the front panel, look under the area "Bias Voltage". The output voltage is switched on using the "ON" button. The other buttons are used to set the voltage. The user should use a voltage multimeter to check the collector voltage by probing the contact labeled TEST. Check the **check step 1** box when the LNA is set.

NOTE Before proceeding to the next step, make sure that the input filter has had enough time to be fully charged so that the device is biased at the chosen operating point.

Step 2 Set the current offset in the LNA. The suggested value is the value measured during the DC trace measurements for that bias point. For the Stanford Research LNA, this has to be done manually. Refer to the area labeled "Input Offset" in the LNA front panel. When set, check the **check step 2** box. Choose a value as near as possible to the DC device output current.

Step 3 Set the sensitivity of the LNA. Refer to the area labeled "Sensitivity" in the LNA front panel. When in Demo mode, the suggested value of 4e–6 Amps/Volt is automatically generated and it is based on the device output conductance. In order to obtain the best performance out of the LNA, the sensitivity of the LNA should be set equal to the device output conductance. However, keep in mind the following considerations:

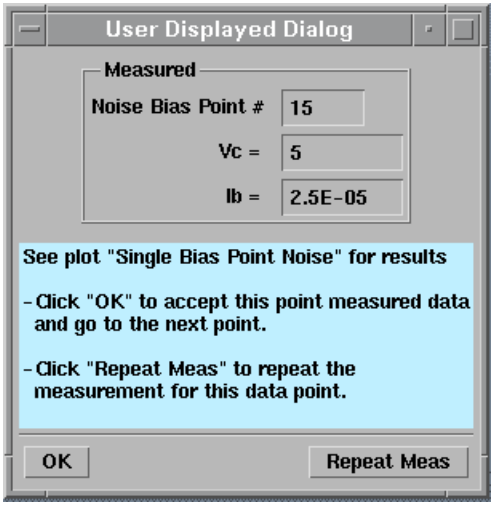
- The output conductance is calculated using a derivative function and therefore may be rather noisy and give non- meaningful values for the sensitivity.
- If the gain is too high, the amplifier output will saturate.

The operator should choose the minimum sensitivity which does not saturate the amplifier output. The operator has to set the actual value in the instrument (in this example: 200e-6 Amps/Volt) and then enter it manually into the box labeled Type Actual Value. When set, click the **check step 3** box.

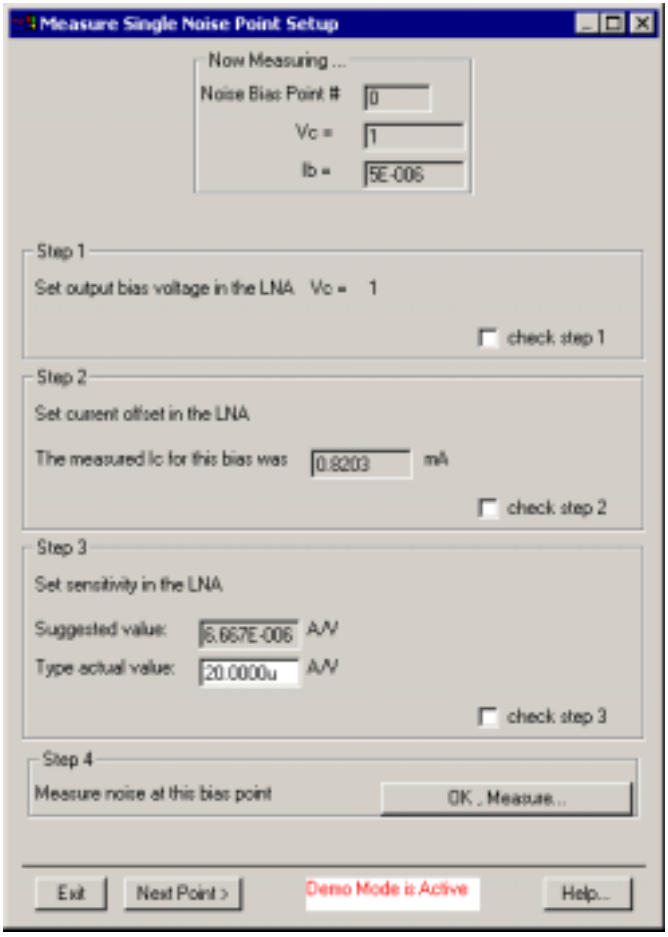
Step 4 Measure noise at this bias point. Click on the button labeled **OK**, Measure to start the noise measurement at this point. After the noise measurement, the Single Point setup will take you automatically to the next point in the sequence.

Instead of measuring the noise at this bias point you can click on **Next Point**. When skipping the point, the noise data in the repository are retrieved. Exit will skip all the remaining points, and end the measurement procedure.

At the end of each Noise point measurement the following dialog pops up:


User Displayed Dialog
Measured
Noise Bias Point #
15
Vc =
5
Ib =
2.5E-05
See plot "Single Bias Point Noise" for results
- Click "OK" to accept this point measured data and go to the next point.
- Click "Repeat Meas" to repeat the measurement for this data point.
OK
Repeat Meas

By clicking **OK**, the operator accepts the measurement data at this bias point and proceeds to the next point in the sequence. Clicking **Repeat Measurements** will display the Single Point measurement dialog to repeat the measurements. The user might vary the LNA settings, the number of averages and repeat the measurement.

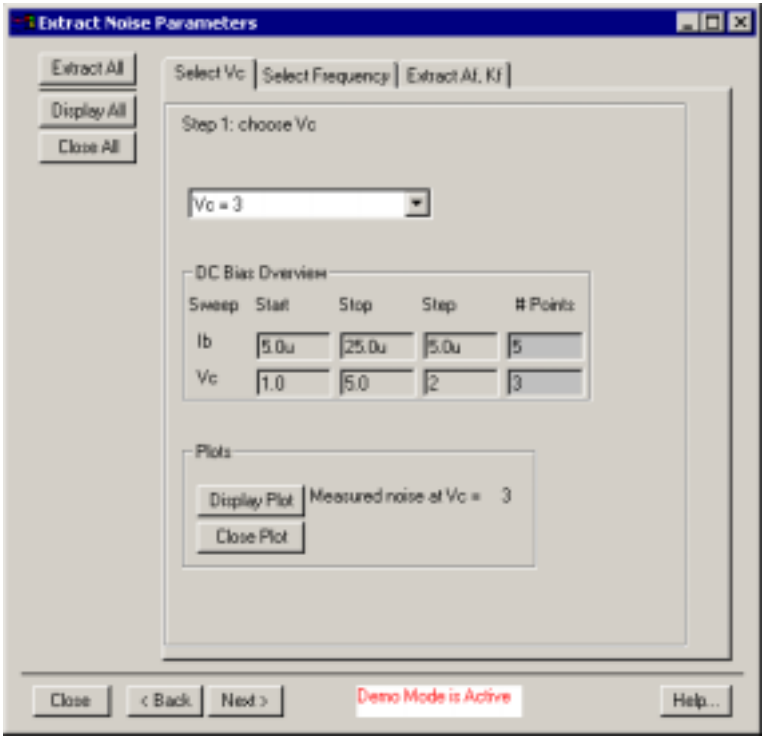


Buttons on the left:

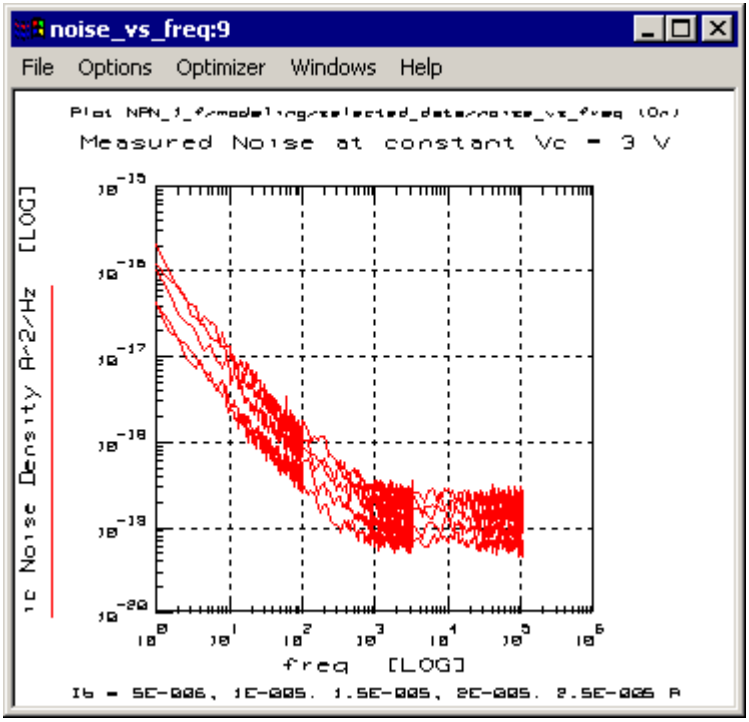
- Measure All: Measures all noise points (Same as Measure All in the Noise Measurement All Points area.
- Display Plots: Displays noise at the selected Vc and the noise of the last point measured.
- Close Plots: Closes all plots.

Start > Extract > Select Vc

Clicking on **Extract** in the Start Wizard window launches the Extract Noise Parameters window shown below. The DC Bias Overview shows the DC setting used for the noise measurement. The user is not allowed to change this table since the noise bias point were set before the Noise Measurements and the noise have been measured already.

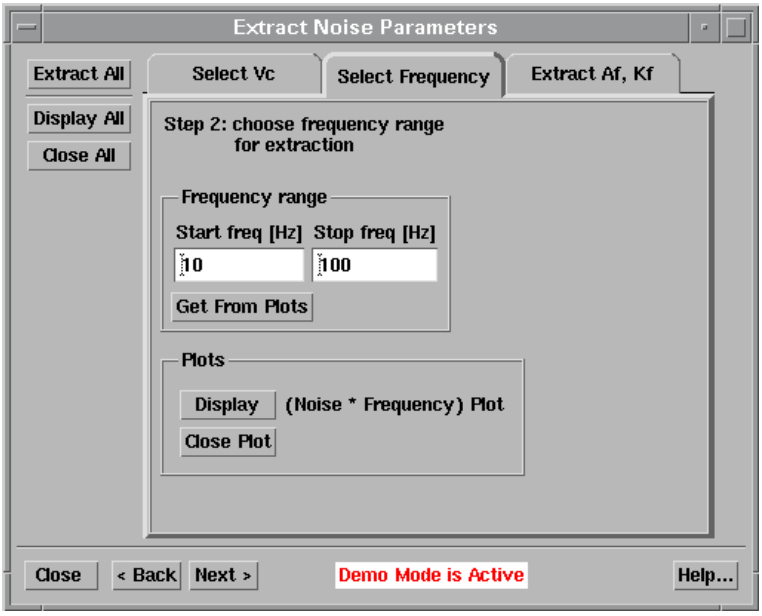


Click on the **Select Vc** tab. Select the Vc value at which you want to perform the extraction. A noise data plot at the selected Vc will be displayed automatically, as shown below.



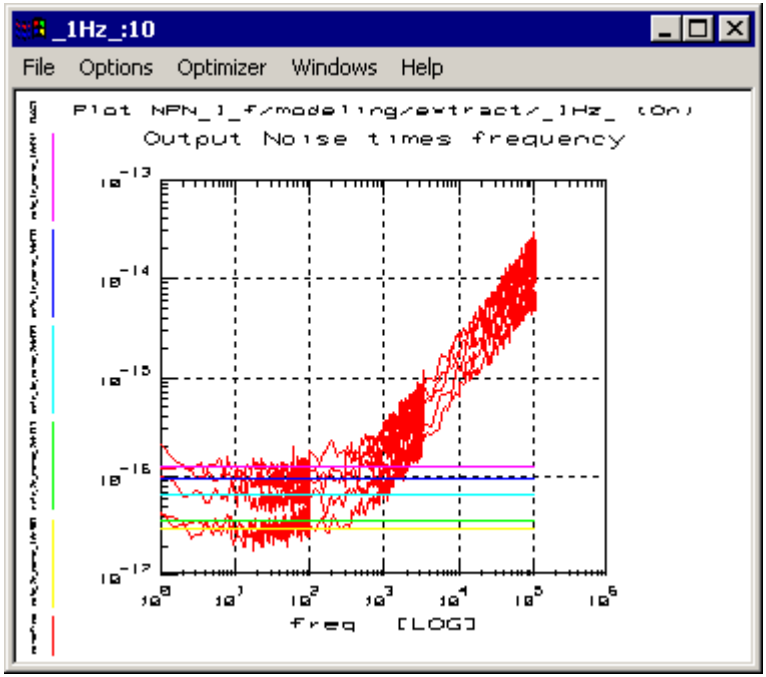
Extraction of noise parameters Af and Kf

Start > Extract > Select Frequency



Click on the **Select Frequency** tab. A Noise · Frequency versus Frequency plot will be displayed automatically, as shown below. On the plot select the area where you would like to extract the noise parameters with a box, then press **Get from Plots**. Alternatively, type the frequency range directly in the Start freq and Stop freq boxes.

NOTE Choose a frequency range where the traces are flat, i.e show an ideal 1/f behavior.

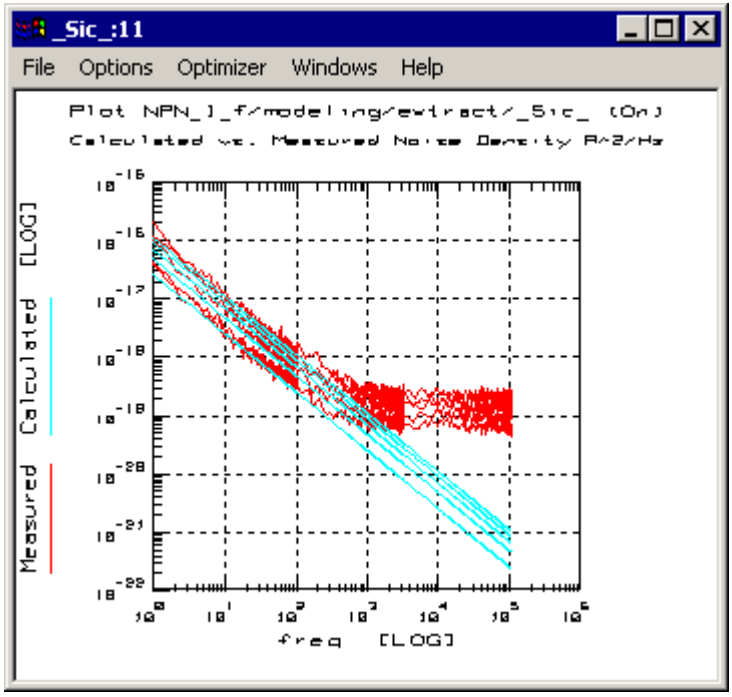


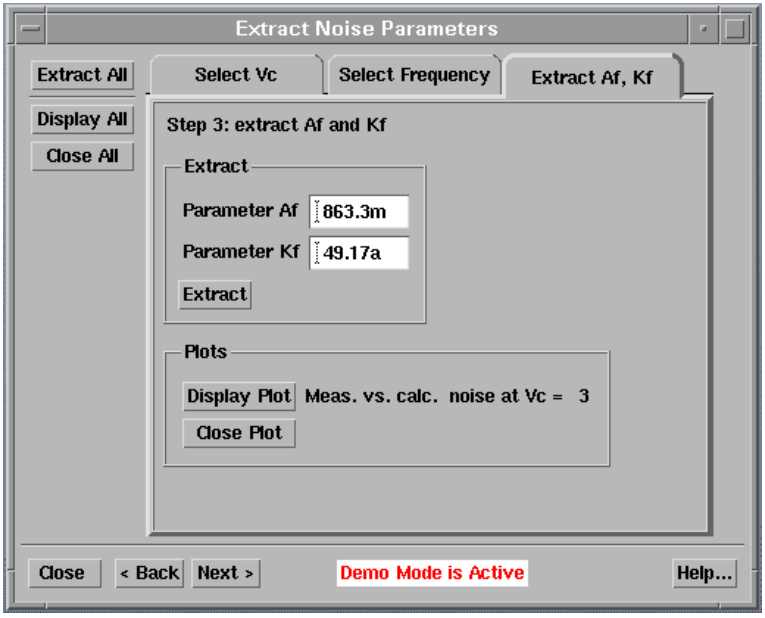
Start > Extract > Extract Af, Kf

Click on the **Extract Af, Kf** tab. Select **Extract** to extract Af and Kf in the selected frequency range. The plot of measured and calculated noise versus frequency, shown below, will appear and the parameters Af and Kf will be extracted.

NOTE The calculated noise sample uses equation (1) for bipolar or (3) for MOS. The calculated noise is only valid in the 1/f noise region.

You can change the values of Af and Kf and the calculated noise will update automatically.


Sic:11
File
Options
Optimizer
Windows
Help
Plot NPN_1_f/modeling/extract/_Sic_ (On)
Calculated vs. Measured Noise Density A^2/Hz
Measured Calculated [LOG]
freq [LOG]



Buttons on Left Side of Dialog Box

- Extract All: automatically extracts Af and Kf using the previously specified Vc and frequency range. Automatically opens all plots and ends in the last tab displaying the extracted Af and Kf.
- Display All: Displays all plots.
- Close All: Closes all plots.

Start > Simulate

Clicking on **SImulate** in the Start Wizard window launches the Simulate Noise window shown below.

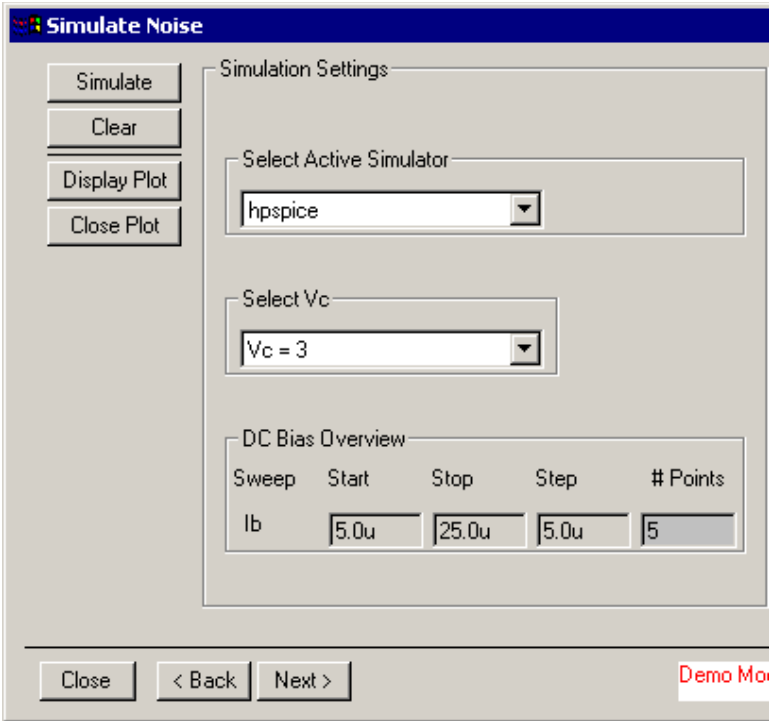


Figure 198 The Simulate Noise Window - LEFT HALF

The simulation dialog has three main areas:

- Simulation settings. Select or change the simulator. Select the collector voltage Vc. Selecting Vc will retrieve the Noise Data from the dataset and display them.
- Tune Noise parameters. Tune parameters Af and Kf.
- Plots. Use plot annotation to add relevant information to simulated Vs measured noise data plot. Press **Update Plots Annotation** to update plot with the text.

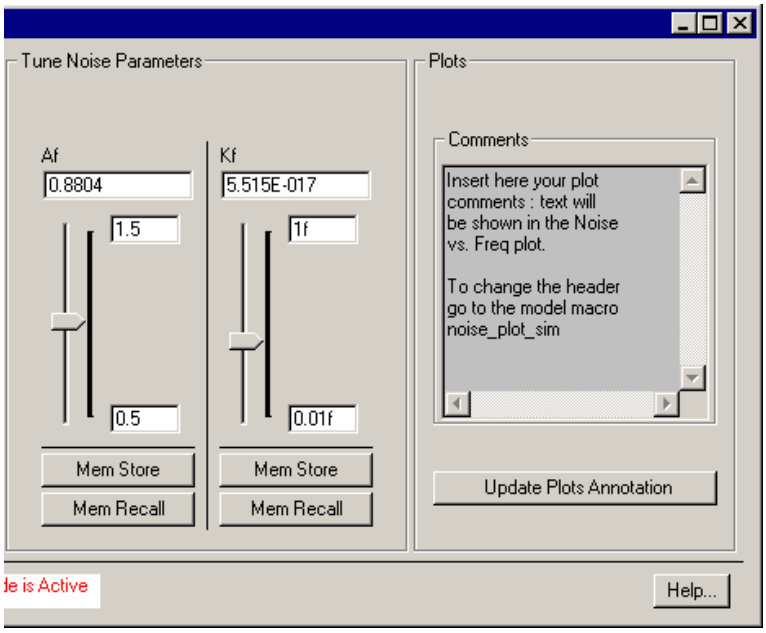
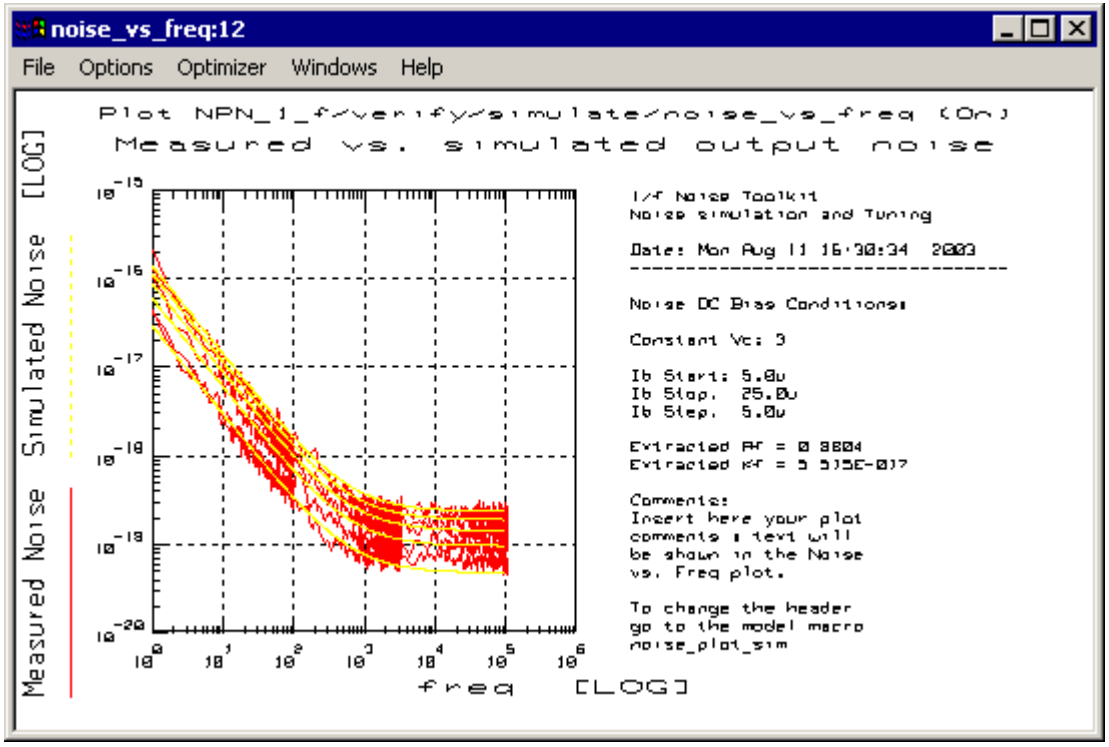


Figure 199 The Simulate Noise Window - RIGHT HALF

The Tune Noise Parameters box

In order to simulate noise values it is necessary to have values of Af and Kf that are a good fit over the measured range of data. This is done by setting their maximum and minimum values - in this example 2.0 and 0.5 for Af, and 1f and 0.01f for Kf. The actual values used are controlled by the slider. In the noise_vs_frequency plot, shown below, the yellow simulated (calculated) curves yellow respond to the slider-selected value of Af, and match the red (measured) data curves best with Af at 979.4m. This “eye-balling” method of curve-fitting works quickly and well.

NOTE The simulation is performed using the selected Vc, change Vc and re-simulate to see how the extracted parameters fit the other Vc values.



Buttons on the Left Side of the Dialog:

Simulate: runs simulation in setup verify/simulate.

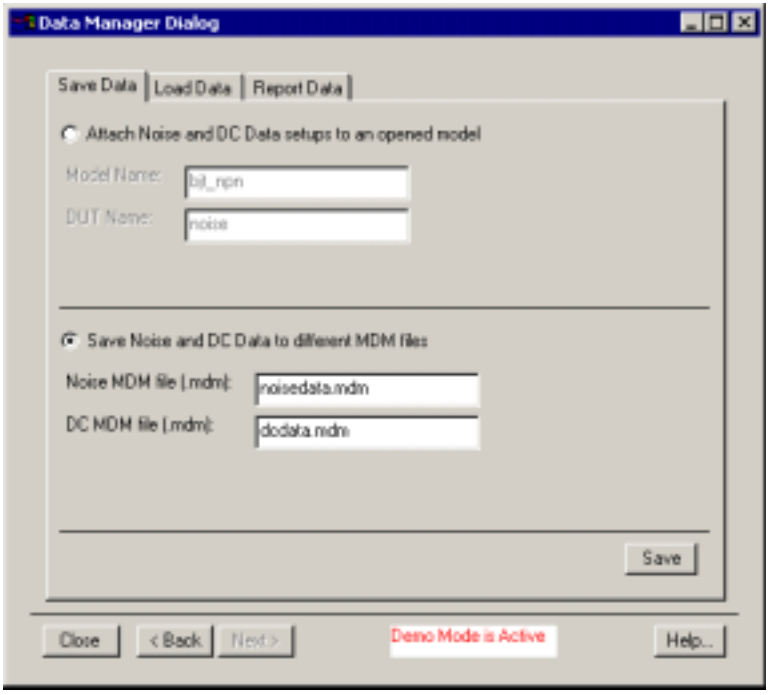
Clear: clears simulated data.

Display Plots: displays measured vs simulated noise data at constant Vc.

Close Plots: closes noise plot.

Start > Data Manager

Clicking on **Data Manager** in the Start Wizard window launches the Data Manager Dialog window shown below.



The Data Manager Dialog has three tabbed folders, titled:

- Save Data
- Load Data
- Report Data Start > Data Manager > Save Data

Click on the **Save Data** tab.

The dialog presents two choices:

Attach Noise and DC Data setups to an opened model

or

Save Noise and DC Data to different MDM files.

Selecting Attach Noise and DC Data setups... allows you to save the noise and DC measurement data to a model. The model must be opened with its icon in the IC-CAP main window. This

option may be useful when you want to keep all the measurement data (electrical, noise, etc.) in the same model file.

Selecting Save Noise and DC Data to different MDM files Noise and DC data are saved in the specified MDM files.

Important Note: Along with the DC and Noise Data, IC-CAP also saves to a DUT or the MDM files some key setting variables and the extracted noise parameters. IC-CAP saves the following variables:

- the DC sweeps variables (inDC_IB_START, inDC_IB_STOP, inDC_IB_STEP, inDC_VCE_START, etc.) and the extracted BF.
- the noise sweep variables (inNOISE_IB_START, inNOISE_IB_STOP, etc.).
- the extracted noise parameters Af, Kf (Ef for CMOS).
- the sensitivity settings of the LNA (Sensitivity). This is essential to display the noise when reading the files back.

Start > Data Manager > Load Data

Click on the **Load Data** folder.

Select **Read Noise and DC Data from an opened model** to read noise and DC measurement data from a model. The model must be opened and located in the IC-CAP main window. Selecting **Read** reads DC and Noise measurement data from the DUT to the repository's setups.

DC_Sweep and Noise Data.

CAUTION Bias settings for DC and Noise measurements will be changed according to the settings stored in the DUT. Actual DC and Noise bias settings, extracted values, and LNA sensitivity settings will be lost and replaced by the values saved in the DUT.

Select **Read Noise and DC Data from MDM files** to save Noise and DC data to the specified MDM files.

CAUTION Bias settings for DC and Noise measurements will be changed according to the settings in the MDM files. Actual DC and Noise bias settings, extracted values and LNA sensitivity settings will be lost and replaced by the values saved in the MDM files.

Start > Data Manager > Report Data

Report Data tab folder.

Select **Generate Report** to write a text report to the specified file name. To change the report format see transform Gui Driver/Data Manager/Generate Text.

1/f Noise Extraction with MOS Transistors

All previous examples in this chapter have dealt with bipolar junction transistors.

The model selected was

```
.../examples/model_files/noise/1_f_toolkit/bjt_1f_noise.mdl.
3For MOS transistors use this model instead
.../examples/model_files/noise/1_f_toolkit/mos_1f_noise.mdl.
```

The basic operation for MOS is the same as for bipolar, except where shown in this section.

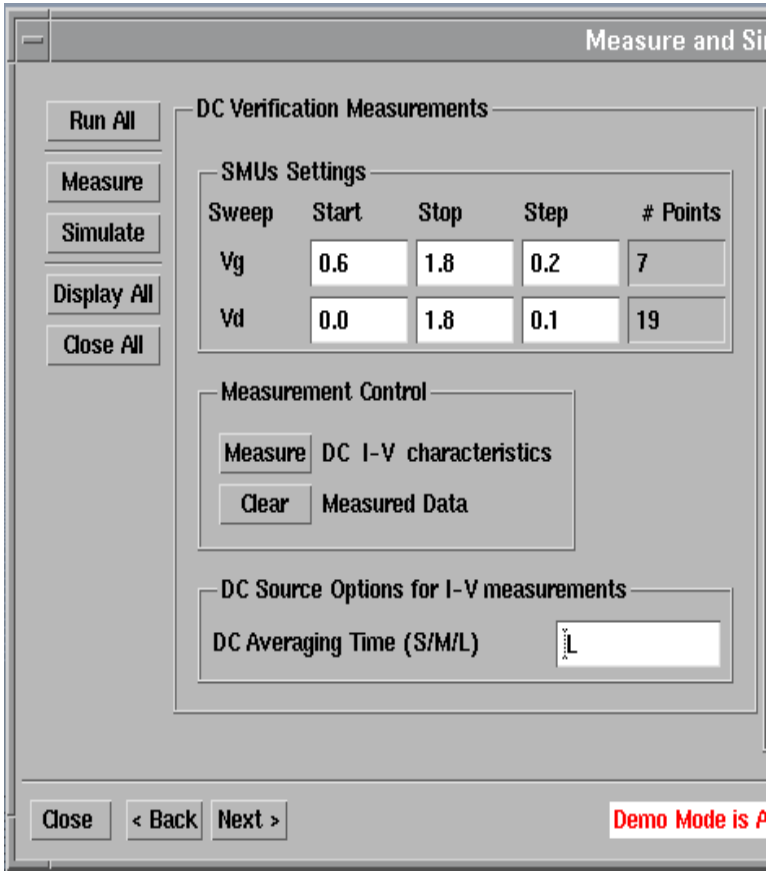
Some Basic Differences

Terminals The emitter, base and collector terminal are replaced by the source, gate and drain terminals respectively.

Impedances The bipolar transistor has a low input impedance from base to emitter, whereas the gate to source impedance of the MOS transistor is essentially infinite.

Currents The drain current Ids replaces the collector current Ice.

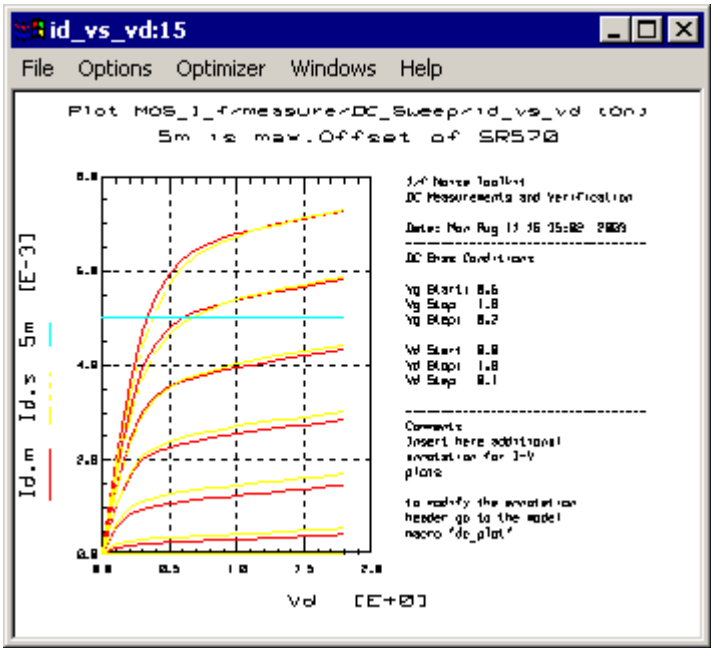
Voltages The voltages Ib and Vce are replaced by Vgs and Vds respectively, as shown in the Measure and Simulate DC Data Dialog Box show below.



Left Side of Measure and Simulate Dialog Box for MOS Model
Note SMU Settings are for Vg and Vd

Figure 200 LEFT SIDE of Measure and Simulate DC Data Dialog Box

Three plots are displayed when the Measure and Simulate DC Data Dialog box is launched. The first is Id versus Vd, as shown below. The second plot is gds versus Vd and the third is ig versus Vd.



The left side of the Measure Noise Data dialog box is shown below. The difference is that the Noise Bias SMUs Settings are for Vg and Vd, rather than Ib and Vc.

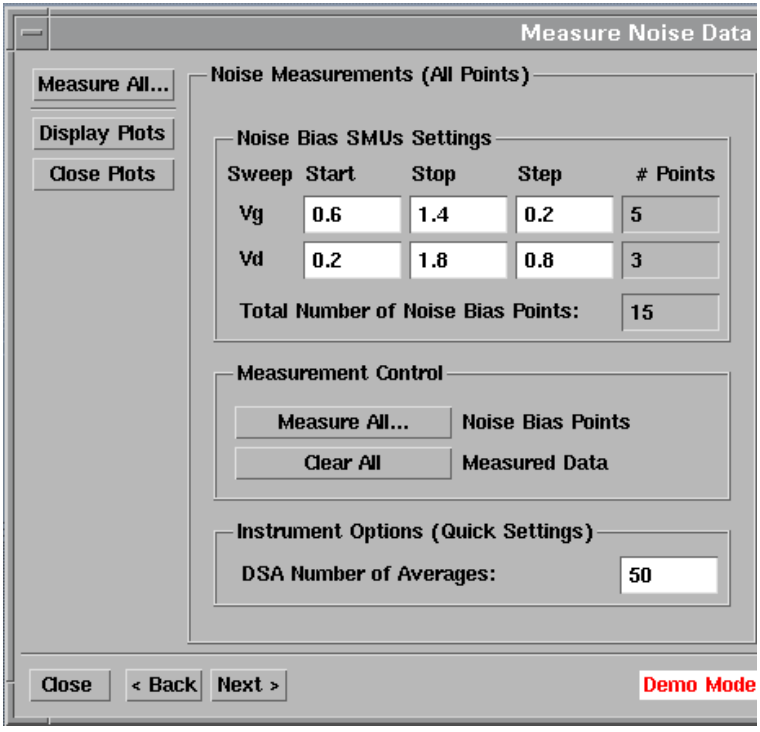


Figure 201 LEFT SIDE of Measure Noise Data Dialog Box
Note SMUs Settings are for Vg and Vd

The left side of the Measure Noise Data dialog box is shown below. The difference is that Choose Vd replaces Choose Vc.

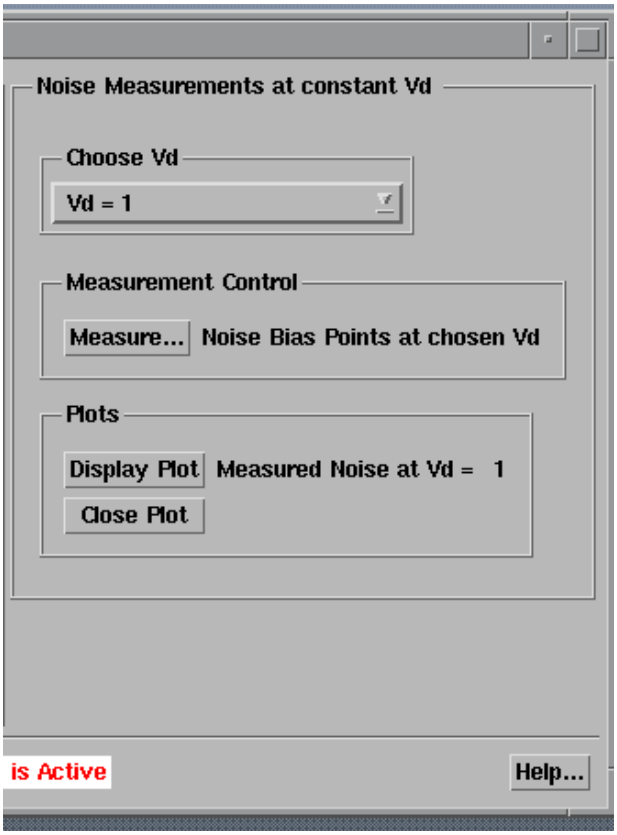
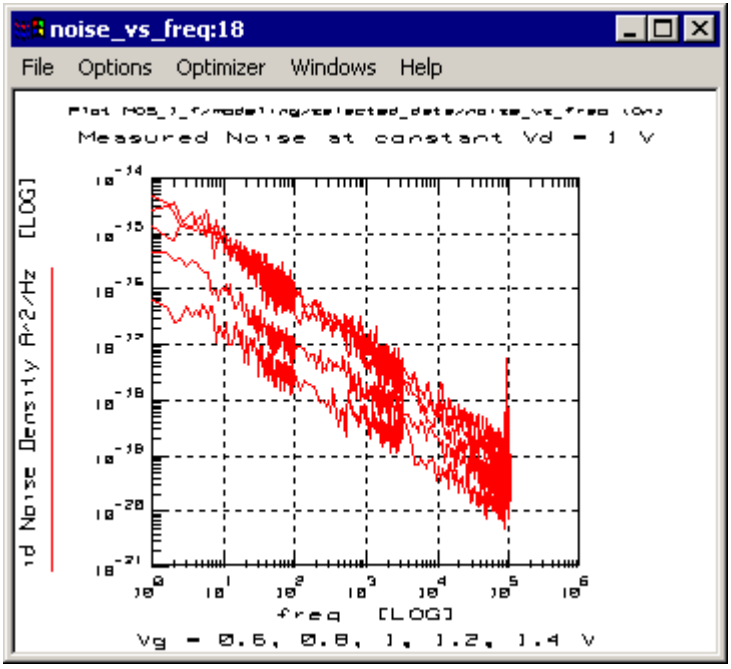


Figure 202 RIGHT SIDE of Measure Noise Data
Choose Vd replaces Choose Vc

The Noise versus Frequency plot shown below is displayed in the Demo Mode when the Measure Noise Data dialog box is launched.



When the Extract Noise Parameters dialog box is launched the previous plot is displayed (also when the Select Vd tab is selected).

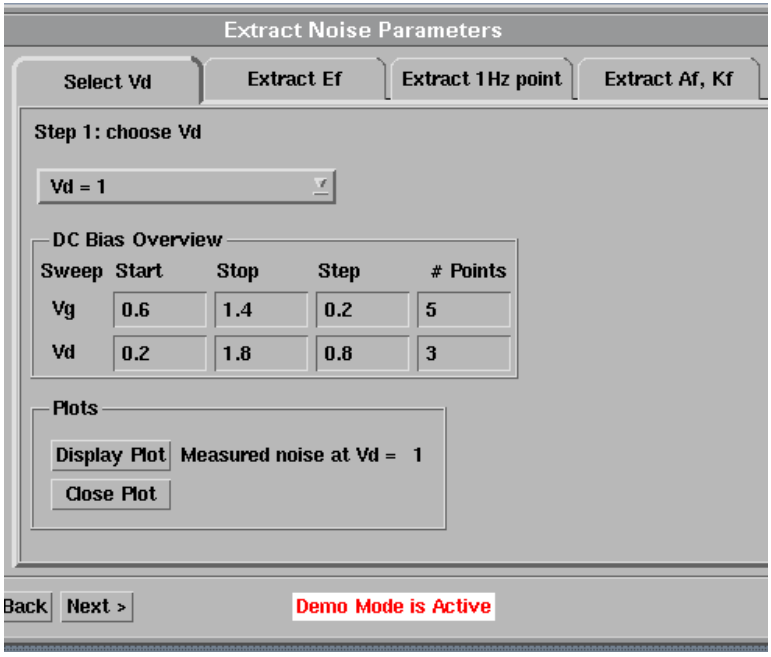
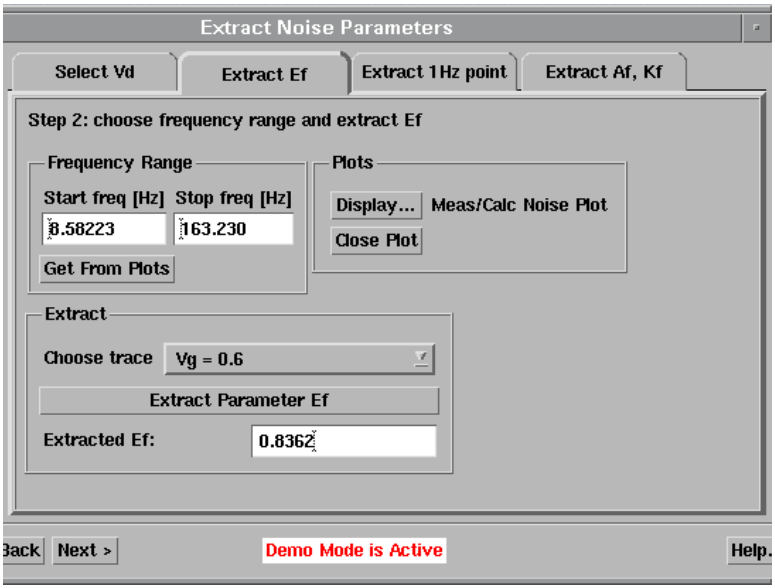
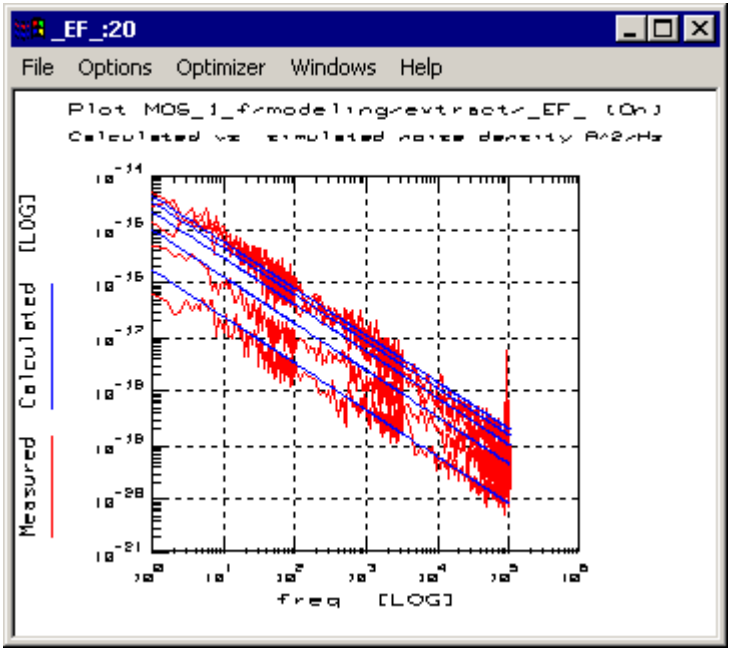


Figure 203 With Select Vd tab selected the DC Bias Overview is for Vg and Vd.

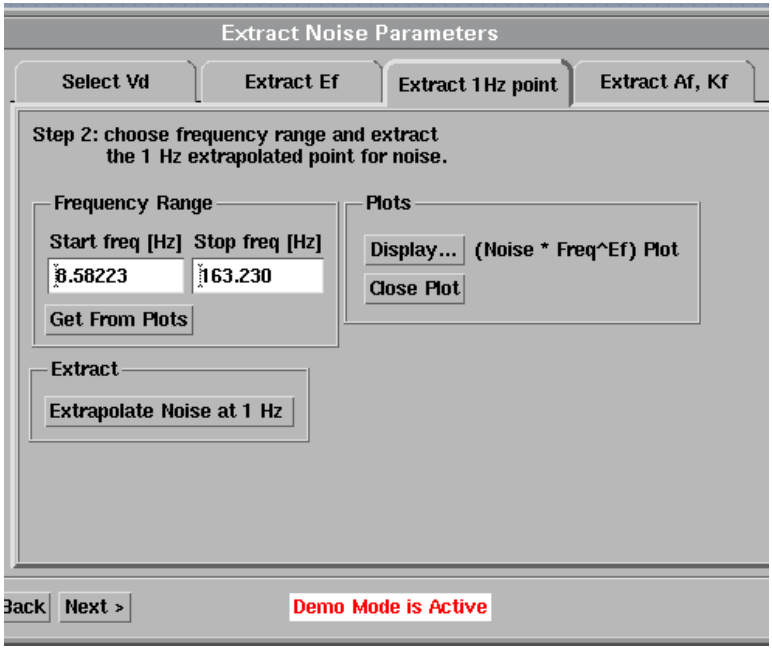
Select the tab **Extract Ef**. A plot showing measured versus calculated noise will be shown. Choose the frequency range in which you want to perform the Ef extraction. The extraction of the Ef is performed using only one of the noise traces. The Extract box is used to pick the Vg trace to be used, as shown below. Choose a trace and click on **Extract Parameter Ef**.



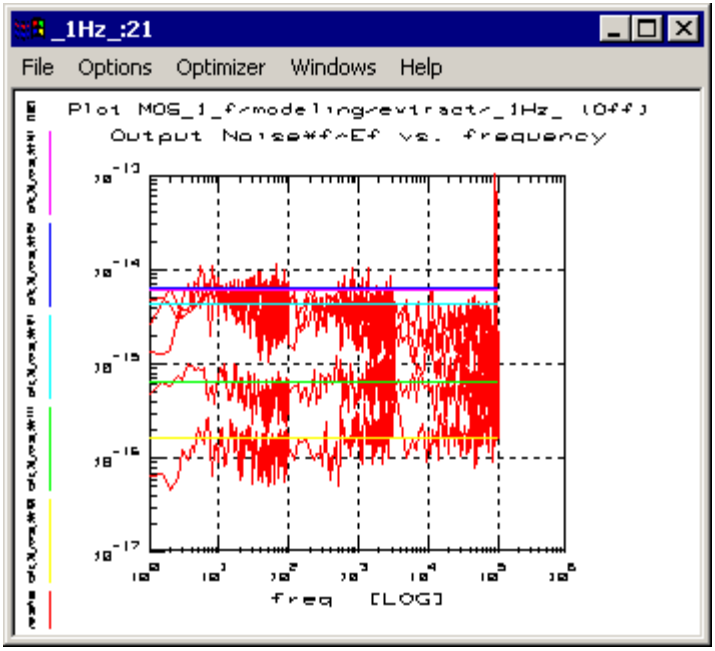
When the **Extract Ef** tab is selected, the Calculated and Simulated values of Ef versus frequency are displayed, as shown below. So far, only the slopes should match—the y-intersects will be modeled by AF and KF in the next steps.



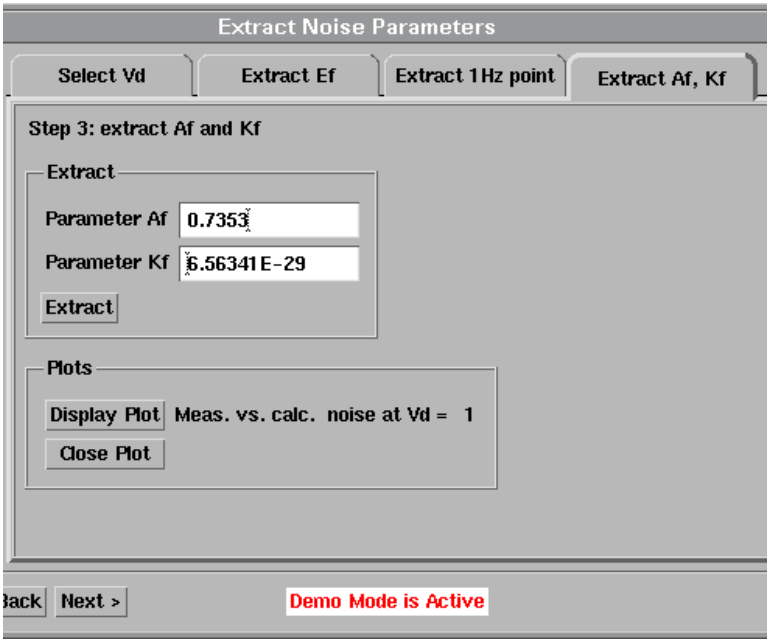
Selecting the **Extract 1Hz Point** tab produces the page shown below. The plot Noise*freq^Ef vs. frequency is shown. Select the frequency range in which you want to extract the parameters Af and Kf. Choose a range for which the traces are flat.

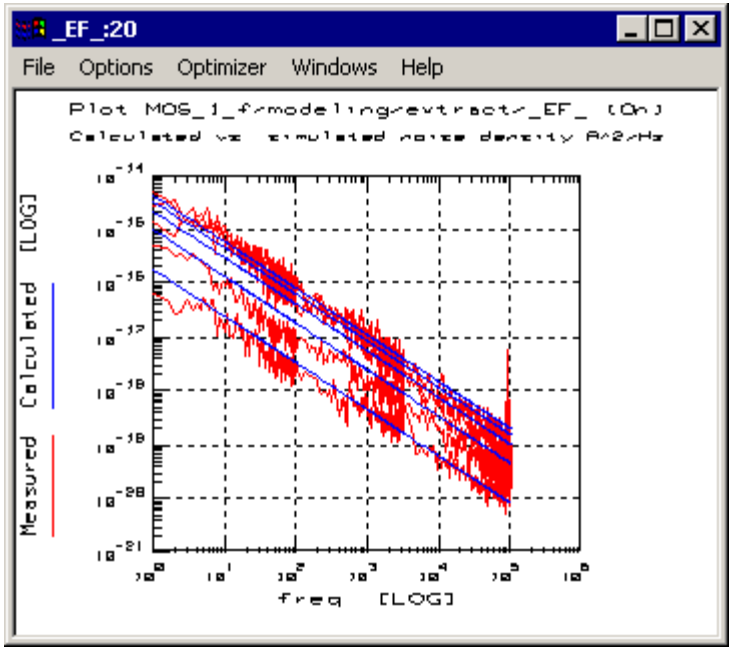


From the Extract 1Hz page the type of plot shown below is produced.

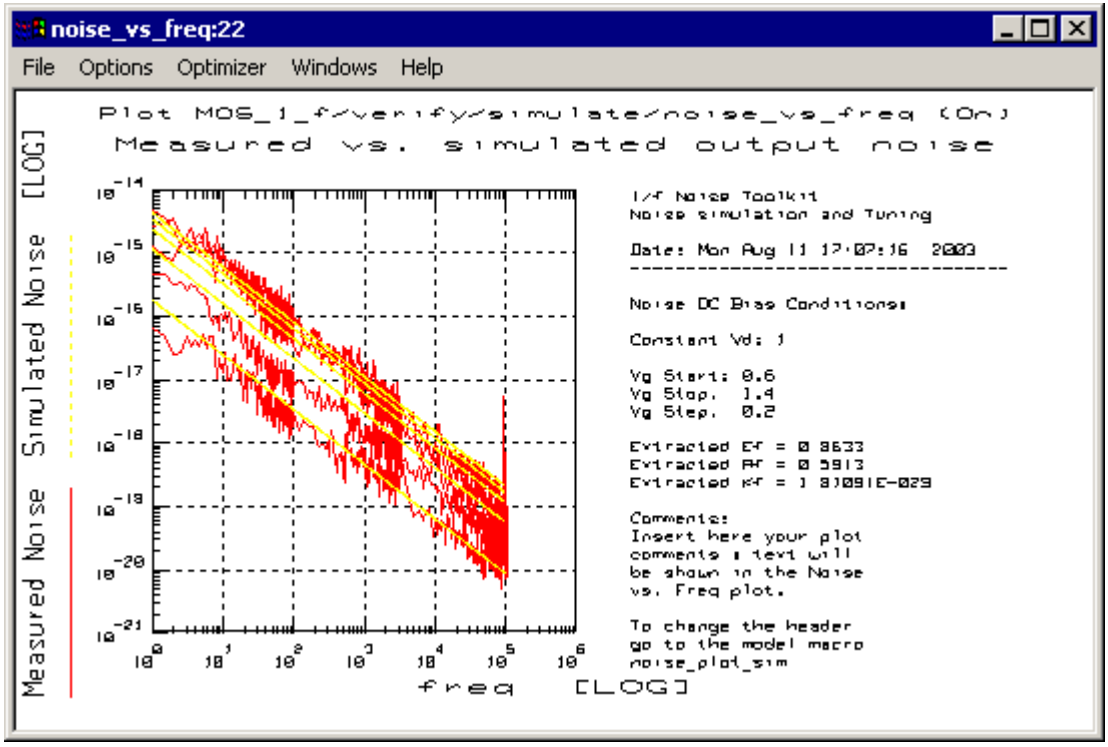


From the Extract Af, Kf page, shown below, a plot of Measured and Calculated Noise versus frequency is produced (shown following dialog box). Click on **Extract** to extract the parameters Af and Kf in the previously selected frequency range.


Extract Noise Parameters
Select Vd
Extract Ef
Extract 1Hz point
Extract Af, Kf
Step 3: extract Af and Kf
Extract
Parameter Af
0.7353
Parameter Kf
6.56341E-29
Extract
Plots
Display Plot
Meas. vs. calc. noise at Vd = 1
Close Plot
ack
Next >
Demo Mode is Active



From the Simulate Noise dialog box a plot of Measured and Simulated Noise versus frequency is produced, as shown below.



1/f Noise Function Descriptions

Function Name	Function Description
Noise_1f_bjt_1Hz	For each noise trace this function finds the 1 Hz intercept point by calculating the average noise in the specified frequency range. The frequency range can be specified by using the variables X_LOW and X_HIGH. If N is the number of noise traces, the function returns an output dataset of size N filled with the N intercept points (one for each trace).

Function Name	Function Description
Noise_1f_bjt_calc	This function calculates the current noise density at the device output (collector) given the frequency range, the device current gain Bf, the base current and the parameters Af and Kf listed in the Parameters table. The noise is calculated as follows: <Sic>= Kf * (ib^Af)/f * Beta^2 If N is the number of traces (number of DC bias points) the inputs are defined as follows: Inputs: Beta: Dc current gain dataset. Size: N frequency: Frequency point dataset. Size: N*freqpoints. Base Current: Base current dataset. Size: N. Output: Dataset filled with the calculated noise. Size: N*freqpoints. Examples: This transform is used during 1/f noise parameters extraction for bipolar devices. See model file examples/model_files/noise/1_f_toolkit/bjt_1f_noise.mdl The transform is used in the setup modeling/extract.
Noise_1f_bjt_extract	This function extracts the parameters Af and Kf. If N is the number of noise traces at a given Vc, inputs and output are as follows: Inputs: Beta: Dc current gain dataset. Size: N. Ic noise 1 Hz: 1 Hz intercept dataset. Size: N. Base Current: base current at each bias point. Size: N. Output: Return a dataset of size N with the calculated 1 Hz values using the extracted Af and Kf. Examples: This transform is used during 1/f noise parameters extraction for bipolar devices. See model file examples/model_files/noise/1_f_toolkit/bjt_1f_noise.mdl The transform is used in the setup modeling/extract.

Function Name	Function Description
Noise_1f_force_bias	This function forces a current or a voltage from the specified unit of a 4142B or 4156B/C. Warning: the instrument will continue to force the bias until the function NOISE_1f_stop_bias is called. Variables: GPIB Address: instrument address. Compliance: voltage or current compliance. Value: voltage or current value to be forced. Parameters: Bias source: specify DC Bias Source Type (4142/4156). GPIB Interface: interface name. Unit Slot (4142) or SMU (4156). Force Current (I) or Voltage (V). Examples: re = NOISE_1f_force_bias(29, 2, 25e-6, “4142”, “hpib”, “2”, “I”). This forces 25 uA of current from unit source on slot 2 of the 4142 at address 29. The interface name is “hpib” and the voltage compliance is 2 V. This transform is used during 1/f noise parameters extraction for bipolar and MOS devices. See model file examples/model_files/noise/1_f_toolkit/bjt_1f_noise.mdl The transform is used in the setup measure/Noise. It is called by the GUI interface function btMeasure located in the setup GuiDriver/MeasureNoise.
Noise_1f_get_Af	This function returns the value of the parameter Af/AF/af stored in the parameter list. Syntax: x = NOISE_1f_get_Af()
Noise_1f_get_Bf	This function returns the value of the parameter Bf/BF/bf stored in the parameter list. Syntax: x = NOISE_1f_get_Bf()
Noise_1f_get_Ef	This function returns the value of the parameter Ef/EF/Ef stored in the parameter list. Syntax: x = NOISE_1f_get_Ef()
Noise_1f_get_Kf	This function returns the value of the parameter Kf/KF/kf stored in the parameter list. Syntax: x = NOISE_1f_get_Kf()
Noise_1f_mos_1hz	For each noise trace this function calculates the 1 Hz intercept point by calculating the average noise in the specified frequency range. The frequency range can be specified by using the variables X_LOW and X_HIGH. If N is the number of noise traces, the function returns an output dataset of size N filled with the N intercept points (one for each trace).
Noise_1f_set_Af	This function sets the value of the parameter Af/AF/af in the parameter list. Syntax: NOISE_1f_set_Af(value)

Function Name	Function Description
Noise_1f_set_Bf	This function sets the value of the parameter Bf/BF/bf in the parameter list. Syntax: NOISE_1f_set_Bf(value)
Noise_1f_set_Ef	This function sets the value of the parameter Ef/EF/ef in the parameter list. Syntax: NOISE_1f_set_Ef(value)
Noise_1f_set_Kf	This function sets the value of the parameter Kf/KF/kf in the parameter list. Syntax: NOISE_1f_set_Kf(value)
Noise_1f_stop_bias	This function stops the bias from the specified DC source. It is used in conjunction with the NOISE_1f_force_bias. Variables: GP-IB Address: instrument address. Parameters: Bias source: specify DC Bias Source Type (4142/4156). GPIB Interface: interface name. Unit Slot (4142) or SMU (4156). Examples: ret = NOISE_1f_force_bias(29, “4142”, “hpib”, “2”) the SMU unit at slot 2 of the 4142 at address 29 will stop any voltage or current bias. This transform is used during 1/f noise parameters extraction for bipolar and MOS devices. See model file examples/model_files/noise/1_f_toolkit/bjt_1f_noise.mdl The transform is used in the setup measure/Noise. It is called by the GUI interface function btMeasure located in the setup GuiDriver/MeasureNoise.

References

1 “Simulating Noise in Nonlinear Circuits Using the HP Microwave and Design System”. Agilent Product Note 85150-4,1993. http://www.agilent.com

2 IC-CAP Noise Modeling Workshop. Agilent Technologies, Germany and Technical University of Munich, Institute of High Frequency Techniques. Munich, Germany 1998.

3 F. X. Sinnesbichler, M. Fisher, G. R. Olbrich, “Accurate Extraction Method for 1/f Noise Parameters

4 Used in G-P Type Bipolar Junction Transistor Models”. IEEE MTT-S Symposium, Baltimore, 1998.

5 C. G. Jakobson, I. Bloom, Y. Nemirovsky. “1/f Noise in CMOS Transistors for Analog Applications”. 19. Convention of Electrical and Electronics Engineers in Israel, 1996. pp. 557-560.

6 Private communication: G. Knoblinger, Infineon AG, Munich and F. Sischka.

7 A. Blaum, O. Pilloud, G. Scalea, J. Victory, F. Sischka. “A New Robust On-Wafer 1/f Noise Measurement and Characterization System”. To be published: ICMTS Conference 2001, Kobe, Japan.

8 Stanford Research SR570 Low-Noise Current Preamplifier.

See http://www.srsys.com

9 A. Van Der Ziel. “Noise, Sources, Characterization, Measurement”, Prentice Hall, 1970.

Index

Numerics

1/f Noise
- Af, 605
- Af MOS, 624
- beta tuning, 593
- data manager, 610
- Ef MOS, 620
- extract Af, Kf, 605
- extract noise parameters, 602
- function descriptions, 627
- general settings, 584
- GUI options, 588
- hardware, 578
- Id/Vd curves, 615
- individual noise measurements, 597
- instrument settings, 585
- Kf, 605
- Kf MOS, 624
- measure noise, 595
- modeling, 570
- MOS transistors, 614
- notes, 582
- references, 631
- simulate, 607
- software, 578
- system settings, 582
- test setup, 595
- tune noise parameters, 609
- verify DC, 590

A

AC extraction, 35
Af MOS, 624
algorithms
- bipolar transistor, 33
- MOSFET, 61

alternative extraction
- Curtice GaAs MESFET, 467
- UCB GaAs MESFET, 442

B

beta tuning, 593
bipolar transistor
- AC extraction, 35
- algorithms, 33
- characterization, 13
- connections, 23
- cv extraction, 34
- DC extraction, 33
- extracting parameters, 29
- extraction guidelines, 25
- instrument options, 25
- instruments, 23
- measurement hints, 26
- model, 15
- NPN, 13
- optimizing, 28
- parameters, 17
- PNP, 28
- resistance extraction, 34, 35
- simulating, 16, 27

bjt_1f_noise.mdl, 579
bjt_npn.mdl, 13
bjt_pnp.mdl, 13
BSIM4
- Acknowledgements, 425
- Capacitance, 85
- Capacitance DUTs, 108
- Data Structure inside the BSIM4 Modeling Package, 72, 400
- DC Diode, 86
- DC Diode DUTs, 116
- DC Transistor, 83
- DC Transistor DUT's, 92
- Drain/Source – Bulk Diodes for DC Measurements, 121
- Measurement, 77, 400
- Measurement Conditions, 81
- Options, 121
- Project Notes, 80
- Switch Matrix, 88
- Temperature Setup, 87
- Teststructures for Deep Submicron CMOS Processes, 398, 399

C

calibration, MOSFET, 51
capacitance extraction
- Curtice GaAs MESFET, 466
- UCB GaAs MESFET, 441

CGaas1.mdl, 449
CGaas2.mdl, 449
characterization circuits, 537

circuits
  characterizing, 537
  characterizing portions, 543
  defining, 540
  design optimization, 549
  ECL OR/NOR logic gate, 541
  extracting parameters, 545
  measurement, 543
  modeling, 537
  multiple instruments, 543
  NPN with PNP, 555
  operational amplifier, 557
  optimizing parameters, 546
  simulation, 549
  test, 551
  types, 555
connections
  bipolar transistor, 23
  MOSFET, 49
Curtice GaAs MESFET
  alternative extraction method, 467
  capacitance parameter extraction, 466, 472
  DC extraction, 471
  device connections, 458
  diode parameter extraction, 465
  extraction algorithms, 471
  extraction guidelines, 460
  extraction procedure, 464
  extraction procedure overview, 463
  inductance extraction, 471
  inductance parameter extraction, 465
  instrument calibration, 461
  instrument options, 460
  linear parameter extraction, 465
  measuring instruments, 461
  model, 451
  N-channel, 453
  optimizing, 463, 470
  P-channel, 453
  resistance extraction, 471
  resistance parameter extraction, 465
  saturation parameter extraction, 465
  simulating, 462, 469
  simulators, 452
  transistor, 449
cv extraction
  bipolar transistor, 34
  MOSFET, 57, 62

D

data manager, 610
DC extraction
  bipolar transistor, 33
  Curtice GaAs MESFET, 471
  MOSFET, 54, 61
  UCB GaAs MESFET, 446
DC Transistor, 83
design optimization circuits, 549
device connections, 458
diode parameter extraction, 465
dutx, 502

E

ECL OR/NOR logic gate, 541
Ef MOS, 620
example files
  bjt_1f_noise.mdl, 579
  bjt_npn.mdl, 13
  bjt_pnp.mdl, 13
  CGaas1.mdl, 449
  CGaas2.mdl, 449
  hnmos6.mdl, 64
  nmos2.mdl, 40
  nmos3.mdl, 40
  npnwpnp.mdl, 34, 556
  opamp.mdl, 558
  pmos2.mdl, 40
  pmos3.mdl, 40
  UCBGaas.mdl, 429
extract
  Af, 605
  Af, Kf, 605
  Kf, 605
  MM9 DUTs, 489
  noise parameters, 602
extracting parameters
  bipolar transistor, 29
  circuits, 545
  MOSFET, 53
extraction
  algorithms
    Curtice GaAs MESFET, 471
    UCB GaAs MESFET, 446
  bipolar transistor, 25
  guidelines
    Curtice GaAs MESFET, 460
    UCB GaAs MESFET, 436
  HSPICE level 6, 67
  MOSFET, 51
  procedure
    Curtice GaAs MESFET, 463
    MESFET, 439
  UCB GaAs MESFET, 440

F

function descriptions, 627

G

geometry, single, 57
GUI options, 588

H

hierarchy test circuit, 553
hnmos6.mdl, 64
HSPICE
  Level 6 MOSFET model, 64

I

Id/Vd curves, 615
inductance extraction
  Curtice GaAs MESFET, 471
  UCB GaAs MESFET, 446
instrument
  calibration
    Curtice GaAs MESFET, 461
    UCB GaAs MESFET, 437
  options
    bipolar transistor, 25
    Curtice GaAs MESFET, 460
    MOSFET, 51
  settings
    1/f Noise, 585
    UCB GaAs MESFET, 436

instruments
bipolar transistor, 23
MOSFET, 49, 51
multiple, 543
UCB GaAs MESFET, 435

J

JUNCAP model, 519

K

Kf MOS, 624

L

linear parameter extraction, 465

M

macro modeling, 537
macros, MM9, 506
measurement circuit, 543
Measurement of the devices, 103
measuring instruments
Curtice GaAs MESFET, 461
UCB GaAs MESFET, 437
MESFET extraction procedure, 439
MM9
DUTs
dutx, 502
extract, 489
quick_ext, 497
macros, 506
model parameters, 479
model variables, 482
parameter extraction, 511
scaling rules, 513
model configuration, UCB GaAs, 434
model parameters
MM9, 479
UCB GaAs MESFET, 433
model variables, MM9, 482
MOS Model 9
See *MM9*
MOS transistors, 1/f Noise, 614
MOSFET
algorithms, 61
calibration, 51
connections, 49
cv extraction, 57, 62
DC extraction, 54, 61
extracting parameters, 53
extraction guidelines, 51
HSPICE level 6 extraction, 67
HSPICE level 6 measurement, 66
HSPICE level 6 model, 64
HSPICE level 6 optimization, 67
HSPICE level 6 parameters, 64
instrument options, 51
instruments, 49, 51
level 2 configuration, 47
model, 41
optimizing, 53, 60
parameters, 43
simulating, 52, 59
simulators, 42
single geometry, 57
multiple instruments circuits, 543

N

N-channel
Curtice GaAs MESFET, 453
UCB GaAs MESFET, 431
nmos2.mdl, 40
nmos3.mdl, 40
noise, 1/f Noise, 595
NPN with PNP circuits, 555
npnwpnp.mdl, 34, 556

O

opamp macro model, 557
opamp.mdl, 558
operational amplifier circuits, 557
optimization
circuit parameters, 546
HSPICE level 6, 67
optimizing
bipolar transistor, 28
Curtice GaAs MESFET, 463, 470
MOSFET, 53, 60
parameters, 546
UCB GaAs MESFET, 438, 444

P

parameter extraction
MM9, 511
parameters
bipolar transistor, 17
MOSFET, 43
P-channel
Curtice GaAs MESFET, 453
UCB GaAs MESFET, 431
pmos2.mdl, 40
pmos3.mdl, 40

Q

quick_ext, 497

R

resistance extraction
bipolar transistor, 34, 35
Curtice GaAs MESFET, 471
UCB GaAs MESFET, 446
resistance parameter extraction, 465

S

saturation parameter extraction, 465
scaling rules, MM9, 513
setup attributes
UCB GaAs MESFET, 457
simulating
1/f Noise, 607
bipolar transistor, 27
Curtice GaAs MESFET, 462, 469
MOSFET, 52, 59
simulation
circuits, 549
UCB GaAs MESFET, 438, 444
simulators
Curtice GaAs MESFET, 452
MOSFET, 42
UCB bipolar, 16
UCB GaAs MESFET, 430
syntax, test circuit, 551
system settings, 1/f Noise, 582

T

Temperature Setup, 87
test circuit, 551
  hierarchy, 553
  syntax, 551
test instruments, UCB GaAs, 435
tune noise parameters, 1/f Noise, 609

U

UCB bipolar transistor
  See *bipolar transistor*
UCB GaAs MESFET
  alternative extraction, 442
  capacitance extraction, 441, 446
  DC extraction, 446
  extraction, 440
  extraction algorithms, 446
  extraction guidelines, 436
  inductance extraction, 446
  instrument calibration, 437
  instrument connections, 435
  instrument settings, 436
  measuring instruments, 437
  model, 429
  model configuration, 434
  model parameters, 433
  N-channel, 431
  optimizing, 438, 444
  P-channel, 431
  resistance extraction, 446
  setup attributes, 457
  simulation, 438, 444
  simulators, 430
  test instruments, 435
UCB MOSFET model, 41
UCBGaas.mdl, 429